Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет кібербезпеки
(повна назва)

Кафедра інформаційно-мережної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Модернізація процесів перерозподілу навантаження в корпоративній мережі

(тема)

Виконав:
здобувач 2 року навчання,
групи ІМІм-24-2
Катерина ЛАТИШЕВА
(власне ім'я, прізвище)

Спеціальність 172 Електронні комунікації та радіотехніка
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Інформаційно-мережна інженерія
(повна назва освітньої програми)

Керівник доц. Наталія ХАРЧЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ Микола МОСКАЛЕЦЬ.
(підпис) (власне ім'я, прізвище)

2026 р.

Не містить відомостей заборонених до відкритого публікування

Студент _____ / Латишева К.С./

Керівник _____ / Харченко Н.А./

Харківський національний університет радіоелектроніки

Факультет кібербезпеки

Кафедра інформаційно-мережної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 172 Електронні комунікації та радіотехніка

Тип програми освітньо-наукова
(код і повна назва)

Освітня програма інформаційно-мережна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві *Латишевій Катерині Сергіївні*
(прізвище, ім'я, по батькові)

1. Тема роботи Модернізація процесів перерозподілу навантаження в корпоративній мережі

затверджена наказом університету від 23 березня 2026р . №232 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 травня 2026 р.

3. Вихідні дані до роботи
Провести аналіз трафіку, що передається в корпоративних мережах. Дослідити методи Керування трафіком, відзначити їх особливості та роль штучного інтелекту у роботі сучасних моделей керування навантаженням. Запропонувати рішення що дозволить ефективно аналізувати трафік у корпоративній мережі та перерозподіляти його між проміжними вузлами. Провести експериментальне дослідження пропонованого методу, відзначити його ефективність.

4. Перелік питань, що потрібно опрацювати в роботі
Вступ
1. Аналіз та класифікація мережного трафіку в корпоративних мережах
2. Аналіз методів керування трафіком
3. Алгоритм динамічного балансування навантаженням
4. Реалізація методу багаторівневого балансування в корпоративній мережі
Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____

Слайди у форматі Power Point (назва та мета роботи, класифікація трафіку за різними ознаками, класифікація засобів моніторингу та аналізу стану комп'ютерних мереж, загальна схема системи управління інформаційними потоками, метод ідентифікації класу трафіку, класи стратегій балансування навантаження, алгоритм дворівневого балансування, оцінка ефективності алгоритму локального балансування, реалізація адаптивного методу балансування навантаженням, прогнозування, розподіл прогнозу за адаптивним рішенням, порівняльний аналіз ефективності моделей машинного навчання, висновки)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ.	23.03.2026	вик.
2	Підбір літератури за темою роботи	24.03 - 10.04.2026	вик.
3	Аналіз та класифікація мережного трафіку в корпоративних мережах	11.04 - 20.04.2026	вик.
4	Аналіз методів керування трафіком	21.04 - 25.04.2026	вик.
5	Алгоритм динамічного балансування навантаженням	26.04 - 13.05.2026	вик.
6	Реалізація методу багаторівневого балансування в корпоративній мережі	14.05 - 18.05.2026	вик.
7	Оформлення презентаційного матеріалу, підготовка до захисту у ЕК	19.05 - 20.05.2026	вик.

Дата видачі завдання 23 березня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Наталія ХАРЧЕНКО
(підпис) (посада, власне ім'я, прізвище

)

РЕФЕРАТ

Пояснювальна записка: 77 с., 23 рис., 3 додатки, 15 джерел.

Об'єкт дослідження – корпоративні мережі.

Мета роботи – дослідження та модернізація існуючих методів та моделей управління трафіком у корпоративних мережах.

Обґрунтовано актуальність проблеми, проаналізовано класифікацію існуючих методів управління трафіком, здійснено їх порівняльний аналіз за критеріями точності, адаптивності, ресурсоємності та придатності до масштабування. Особливу увагу зосереджено на побудові та впровадженні методу динамічного балансування навантаженням, що поєднує принципи пріоритетного обслуговування з машинним прогнозуванням навантаження. Метод реалізовано в середовищі Google Colaboratory з використанням бібліотек Python. Проведено генерацію синтетичних типів трафіку, побудовано прогнозну модель із використанням алгоритмів машинного навчання (лінійна регресія, дерево рішень, випадковий ліс), виконано аналіз точності та ефективності прийняття рішень щодо обслуговування різних класів даних.

ДИНАМІЧНЕ УПРАВЛІННЯ ТРАФІКОМ, КОРПОРАТИВНА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, QOS, АДАПТИВНІСТЬ, МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯМ.

THE ABSTRACT

Explanatory note: 77 p., 23 fig, 15 sources, 3 app.

The object of research is corporate networks.

The purpose of the work is to research and modernisation of existing traffic management methods and models in corporate networks.

The relevance of the problem is demonstrated, the classification of existing traffic management methods is analysed, and a comparative analysis of these methods is carried out based on criteria of accuracy, adaptability, resource consumption and scalability. Particular attention is focused on the development and implementation of a dynamic load balancing method that combines the principles of priority-based service with machine-based load forecasting. The method has been implemented in the Google Colaboratory environment using Python libraries. Synthetic traffic patterns were generated, a predictive model was built using machine learning algorithms (linear regression, decision tree, random forest), and an analysis of the accuracy and effectiveness of decision-making regarding the handling of different data classes was carried out.

DYNAMIC TRAFFIC MANAGEMENT, CORPORATE NETWORK, MACHINE LEARNING, FORECASTING, QOS, ADAPTIVITY, LOAD BALANCING METHOD.

ЗМІСТ

С.

ПЕРЕЛІК СКОРОЧЕНЬ

ACL – список керування доступом;
AI – штучний інтелект;
ANN – штучна нейронна мережа;
API – програмний інтерфейс застосунку;
CPU – центральний процесор;
CSV – формат табличних даних;
DNX – механізм балансування побудований за моделлю «брокер-агент»;
GNS3 – Graphical Network Simulator 3;
GPU – графічний процесор;
GUI – графічний інтерфейс користувача;
HTML – мова розмітки гіпертексту;
IFS – метод систем ітераційних функцій;
MAE – середня абсолютна похибка;
ML – машинне навчання;
MSE – середньоквадратична похибка;
NFV – віртуалізація мережевих функцій;
NS-3 – Network Simulator версії 3;
OMNeT++ – об’єктно-орієнтоване середовище моделювання мереж;
OPNET – Optimized Network Engineering Tools;
PDU – одиниця передачі даних;
QoS – якість обслуговування;
RMSE – корінь середньоквадратичної похибки;
SLA – угода про рівень обслуговування;
TPU – тензорний процесор;
КМ – комп’ютерна мережа.

ВСТУП

В епоху активної цифрової трансформації бізнесу корпоративні мережі відіграють фундаментальну роль у забезпеченні безперебійної та ефективної діяльності організацій. Збільшення обсягів обміну інформацією, широке впровадження хмарних рішень, поширення віддаленої роботи, активне використання відеоконференцій, а також інтеграція систем моніторингу та управління – все це створює значне зростання навантаження на існуючу мережеву інфраструктуру. Як наслідок, ефективне управління мережевим трафіком стає критично важливим елементом для підтримки високої продуктивності, гарантування надійності та забезпечення безпеки корпоративної мережі.

Неефективне управління трафіком може мати серйозні негативні наслідки: від затримок у передачі даних та втрати пакетів до перевантаження окремих вузлів мережі. Це безпосередньо впливає на якість наданих послуг, призводить до фінансових втрат та завдає шкоди репутації компанії. Тому зростає попит на інноваційні методи та моделі, які б дозволяли не тільки аналізувати поточний стан мережі, але й прогнозувати майбутню поведінку трафіку, динамічно адаптуватися до змін та забезпечувати оптимальний розподіл мережевих ресурсів.

Метою кваліфікаційної роботи є дослідження існуючих методів управління трафіком у корпоративних мережах, а також розробка та експериментальне дослідження методу динамічного балансування із застосуванням прогнозування навантаження, що дозволяє оптимізувати розподіл мережевих ресурсів на основі завчасного виявлення пікових станів.

1 АНАЛІЗ ТА КЛАСИФІКАЦІЯ МЕРЕЖНОГО ТРАФІКУ В КОРПОРАТИВНИХ МЕРЕЖАХ

1.1 Розвиток передачі даних в корпоративних мережах

Корпоративна мережа - це об’єднане середовище, яке інтегрує комп’ютери, сервери, сховища даних, мережеве обладнання, програмні засоби та користувачів в єдину інформаційну систему. Вона забезпечує безперервний зв’язок між усіма підрозділами організації, незалежно від їх розташування, що дає можливість централізовано керувати інформацією, обмінюватися нею, а також отримувати доступ до систем управління ресурсами, звітності та підтримки клієнтів.

Дані, що циркулюють у такій мережі, можуть мати різні формати та рівень структурування. Структуровані дані - це масиви інформації зі строго визначеною формою, наприклад, таблиці або записи в реляційних базах даних. Неструктуровані дані не мають чіткої організації, ними можуть бути електронні листи, текстові файли, аудіо- та відеоматеріали, зображення або публікації в соцмережах. Окремо виділяють напівструктуровані дані, які мають певний порядок, але не відповідають класичним базам даних.

Обробка даних у корпоративному середовищі включає не лише технічні операції, а й етапи збору, відбору, об’єднання, перетворення, аналізу та подальшого подання у зрозумілому вигляді. Важливо враховувати складну систему прав доступу, ролей користувачів, рівнів безпеки та дотримання політик щодо зберігання і опрацювання інформації згідно з нормативними вимогами.

Варто підкреслити, що корпоративні дані не існують самостійно - вони є невід’ємною частиною бізнес-процесів організації, що надає їм зміст і значення. Саме через це виникає потреба у створенні комплексної інфраструктури для їхньої обробки, яка відповідала б розмірам, структурі та швидкості розвитку компанії.

Ключовим елементом корпоративної архітектури є організація розміщення інформації та обчислювальних потужностей, яка може бути як централізованою, так і розподіленою. У централізованих моделях всі основні обчислювальні процеси та збереження даних відбуваються на одному сервері

або в корпоративному дата-центрі, що дає змогу ефективно контролювати доступ, оновлення та резервне копіювання. У розподілених системах обробка та зберігання функцій делегуються багатьом вузлам мережі, що знижує затримки, підвищує гнучкість і дозволяє розміщувати дані ближче до користувачів, особливо коли філії підприємства розташовані у різних географічних точках.

Хмарні технології кардинально змінили підходи до побудови систем. Завдяки віртуалізації, послугам інфраструктури (IaaS) та платформ (PaaS), компанії отримали змогу легко збільшувати обчислювальні потужності, запускати нові сервіси без великих витрат на обладнання та співпрацювати з іншими постачальниками. Дані тепер можуть зберігатися в хмарі, оброблятися на віртуальних кластерах, а аналітика доступна через веб з будь-якої точки світу. Однак, такі рішення потребують ретельної уваги до безпеки, контролю доступу, захисту від витоків та відповідності стандартам зберігання даних [1].

Системи обробки даних включають модулі для підключення до різних джерел: від внутрішніх програм обліку та CRM до сенсорів, веб-сервісів та баз даних партнерів. Далі йдуть сервіси, які трансформують, агрегують, фільтрують та перевіряють інформацію, використовуючи як звичайні інструменти, так і спеціалізовані рішення на кшталт Apache Spark. Важливу роль відіграють інтерфейси, що дозволяють користувачам бачити результати аналітики через візуалізації, звіти, дашборди або API для інтеграції з іншими програмами.

Ефективна система обробки даних має враховувати специфіку галузі, розмір компанії, типи даних та рівень її цифрової зрілості. Тільки гнучка, масштабована та безпечна архітектура дозволить повністю автоматизувати аналітику, знизити витрати та покращити якість управлінських рішень на основі актуальних даних.

У корпоративних мережах успішна обробка даних неможлива без ретельного збору та підготовки. Цей етап визначає не тільки обсяг та актуальність інформації, але й її якість, повноту та придатність для аналізу. Часто саме на цьому етапі дані втрачають свою цінність, що може призвести до помилкових висновків. Тому методи збору та підготовки даних є надзвичайно важливими.

У корпоративному середовищі дані надходять з різнорідних джерел, кожне з яких характеризується унікальною структурою, форматом та частотою оновлення. До ключових джерел належать внутрішні інформаційні системи

підприємства, зокрема ERP-системи, бухгалтерські та логістичні платформи, а також зовнішні веб-сервіси, API-інтерфейси, інструменти моніторингу користувацької поведінки, пристрої інтернету речей та партнерські бази даних. Зважаючи на високу гетерогенність цих джерел, процес збору даних передбачає використання спеціалізованих інтеграційних інструментів, які забезпечують уніфікацію вхідної інформації та її консолідацію в єдиному середовищі обробки [1].

Традиційні корпоративні системи переважно використовують реляційні бази даних та SQL-запити для обробки даних. Цей підхід ефективно підтримує агрегацію, сортування, фільтрацію та вибірку даних за різними критеріями, що є достатнім для більшості базових завдань звітності. Крім того, активно застосовуються OLAP-аналітичні інструменти, що забезпечують багатовимірний аналіз даних у форматі кубів, дозволяючи менеджерам та аналітикам оперативно отримувати уявлення про ключові показники ефективності. У випадках, коли виникає потреба в обробці великих обсягів даних або побудові складних сценаріїв, ці методи доповнюються засобами пакетної обробки, яка виконується за розкладом або за певними тригерними умовами.

Однак, в умовах сучасного цифрового ландшафту зростає актуальність обробки даних у режимі реального часу. На противагу традиційному підходу, що передбачає накопичення та аналіз інформації з часовим лагом, потокова обробка набуває все більшої популярності. Вона дозволяє обробляти події та транзакції миттєво після їх виникнення. Такі системи є критично важливими для таких галузей, як фінансовий сектор, телекомунікації, електронна комерція та кібербезпека, де затримка навіть у кілька секунд може мати значні негативні наслідки. Інструменти, такі як Apache Kafka, Apache Flink або Spark Streaming, забезпечують ефективну реалізацію цих підходів, дозволяючи працювати з потоками даних у великих обсягах та з високою швидкістю [1].

З розвитком технологій великих даних компанії все частіше звертаються до методів, що базуються на архітектурі Hadoop або розподілених обчислювальних фреймворках. Це дозволяє працювати з інформацією, яка не може бути оброблена традиційними засобами через її обсяг, швидкість генерації або різноманітність форматів. Обробка Big Data передбачає не лише технічну здатність обробити великі масиви, але й застосування нових підходів

до зберігання, індексації, паралельної обробки та розподілу навантаження між кластерами.

Зростає значення інтелектуального аналізу даних, який охоплює широкий спектр методів, включаючи машинне навчання, класифікацію, кластеризацію, створення прогнозних моделей, розробку рекомендаційних систем та глибинне навчання. Ці передові підходи дозволяють не просто аналізувати наявну інформацію, а й виявляти неочевидні закономірності, прогнозувати майбутні події та приймати гнучкі рішення в умовах невизначеності. Моделі, здатні навчатися на історичних даних та автоматично адаптуватися до нових вхідних параметрів, є надзвичайно цінними для таких сфер, як маркетинг, персоналізація послуг, оптимізація логістичних ланцюгів та управління ризиками [2].

Важливою складовою ефективної роботи з даними є їхня візуалізація та представлення у форматі, зрозумілому для кінцевих користувачів. Методи візуалізації, такі як інтерактивні дашборди, різноманітні графіки, картографічні представлення та теплові карти (heatmap-аналіз), слугують завершальним етапом обробки. Вони перетворюють результати складних обчислень у наочну форму, що сприяє прийняттю обґрунтованих рішень. Це дає змогу керівництву швидко інтерпретувати отримані дані, оперативно реагувати на зміни в динаміці бізнес-процесів та виявляти критичні точки в режимі реального часу.

У контексті корпоративних мереж, де обсяг, швидкість надходження та критичність даних постійно зростають, питання безпеки інформації виходить за рамки суто технічної проблеми, набуваючи стратегічного значення. Процес обробки даних передбачає їхнє постійне переміщення, трансформацію, зберігання та надання доступу, що створює потенційні вразливості на кожному з цих етапів. Зловмисники можуть використовувати ці слабкі місця для отримання несанкціонованого доступу, викрадення конфіденційної інформації або здійснення маніпуляцій з даними. Це може призвести не лише до значних фінансових втрат, але й до серйозних репутаційних наслідків для організації. Тому забезпечення безпеки даних під час їхньої обробки є обов'язковою умовою для стабільного функціонування будь-якої корпоративної інформаційної системи.

Фундаментом захисту даних є побудова чіткої системи ідентифікації та автентифікації користувачів. Ця система гарантує, що доступ до корпоративних ресурсів мають виключно уповноважені особи. У корпоративних системах для

досягнення цієї мети часто застосовуються багаторівневі механізми перевірки. До них належать одноразові паролі, використання апаратних токенів, біометричні ідентифікатори або інтеграція з централізованими системами управління доступом. Однак, навіть після успішного підтвердження особи, надзвичайно важливим є обмеження повноважень. Це досягається шляхом запровадження політик авторизації, які надають кожному користувачеві доступ лише до тієї частини даних, яка безпосередньо необхідна для виконання його посадових обов'язків.

Сучасна робота з даними неможлива без криптографічних методів захисту. Шифрування даних під час їх передачі по мережі та зберігання в базах даних значно знижує ймовірність несанкціонованого доступу чи витоку. Надійні алгоритми шифрування (симетричні та асиметричні), протоколи безпеки (SSL/TLS), цифрові підписи та механізми перевірки цілісності даних є основою безпечного обміну інформацією в цифровому світі [3]. Це особливо важливо при роботі з даними в хмарі або при обміні інформацією між різними компаніями. Однак, для сенсорних мереж існують альтернативні підходи, які не потребують криптографічного захисту (рис. 1.1).

Рисунок 1.1 – Сучасні методи обробки даних у комп’ютерних мережах

1.2 Різновиди мережевого трафіку

Основоположним елементом телекомунікацій та комп'ютерних мереж є мережевий трафік (рис. 1.2). Він визначається як обсяг даних, що транспортуються мережею протягом певного періоду. Трафік складається з окремих пакетів, кожен з яких несе як корисні дані, так і службову інформацію, необхідну для роботи мережевих протоколів.

Якщо розглядати трафік у корпоративному середовищі, то це сукупність усіх інформаційних потоків, що виникають внаслідок взаємодії користувачів, сервісів та програм у рамках корпоративної інформаційної системи. Розуміння властивостей та складу цього трафіку є критично важливим для оптимізації використання мережевих ресурсів та підтримання високої продуктивності [4].

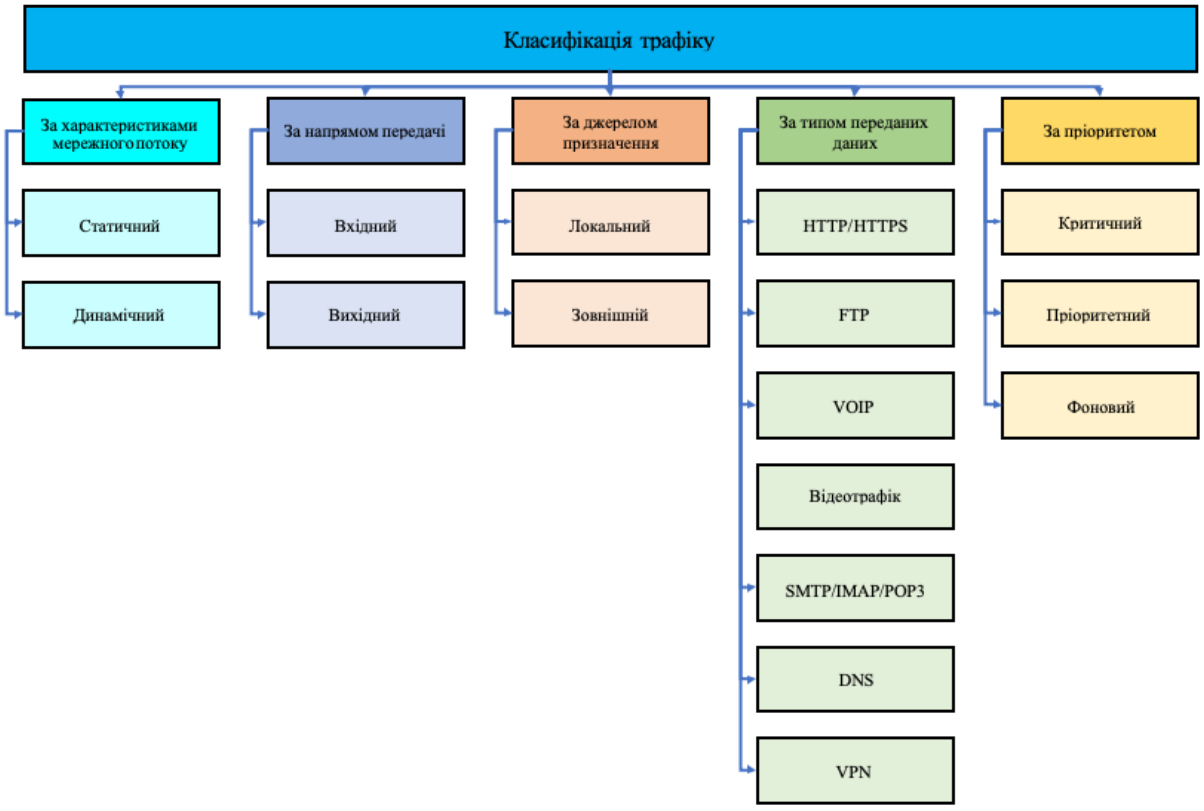

Рисунок 1.2 – Класифікація трафіку за різними ознаками

Для ефективного управління та глибокого розуміння роботи мережі, трафік поділяють на різні категорії за певними ознаками. Ця класифікація допомагає краще аналізувати його поведінку та оптимізувати мережеві ресурси. Нижче наведено основні способи класифікації [4]:

1) за напрямком руху даних:

- вхідний трафік - дані, які надходять до вашої внутрішньої мережі ззовні (наприклад, з Інтернету);
- вихідний трафік - дані, які відправляються з вашої внутрішньої мережі назовні;

2) за місцем походження та призначенням:

- локальний трафік: обмін даними відбувається виключно між пристроями в межах однієї корпоративної мережі;
- зовнішній трафік: дані передаються між вашою мережею та ресурсами в Інтернеті або іншими віддаленими мережами;

3) за типом передаваної інформації:

- HTTP/HTTPS трафік, пов'язаний з переглядом веб-сайтів та обміном даними через браузер;
- FTP використовується для передачі файлів між комп'ютерами;
- VoIP дані для голосового зв'язку через мережу (інтернет-дзвінки);
- Відеопотоки - передача відео в реальному часі (наприклад, з YouTube, Zoom, Teams);
- електронна пошта (SMTP/IMAP/POP3) - трафік, що забезпечує роботу електронної пошти;
- DNS запити до системи доменних імен, яка перетворює назви сайтів на IP-адреси;
- VPN - зашифрований трафік, що передається через захищені тунелі;

4) за рівнем важливості (QoS - Quality of Service):

- критичний трафік: найвищий пріоритет мають дані, які потребують миттєвої доставки (наприклад, VoIP-дзвінки, фінансові транзакції);
- пріоритетний трафік: важливі службові дані, які мають вищий пріоритет, ніж фоновий трафік;
- фоновий трафік: менш термінові передачі даних, такі як оновлення програмного забезпечення або резервне копіювання;

5) за характером передачі даних:

- статичний трафік - передача даних відбувається з відносно постійною швидкістю;
- динамічний трафік - швидкість передачі даних змінюється, можуть спостерігатися періоди високої активності.

Кожен потік даних у мережі має свої специфічні параметри, які визначають його якість та ефективність [4]:

- пропускна здатність - максимальний обсяг даних, який може бути переданий за певний проміжок часу;
- затримка (латентність) - час, необхідний для доставки пакета даних від відправника до отримувача;
- варіативність затримки (джиттер) - різниця в часі доставки між послідовними пакетами даних;
- втрати пакетів - відсоток пакетів даних, які не досягли свого призначення;
- частота передачі - кількість пакетів даних, що надсилаються за секунду.

Моніторинг та управління типами та обсягами трафіку в корпоративній мережі надає значні переваги [4]:

- забезпечення пріоритету для критично важливих сервісів: гарантує стабільну роботу найважливіших додатків та служб;
- виявлення підозрілої активності: допомагає швидко ідентифікувати ознаки кібератак або технічних збоїв;
- планування розвитку мережі: надає дані для обґрунтованого планування розширення та модернізації мережевої інфраструктури;
- реалізація політик безпеки: дозволяє встановлювати правила доступу та контролювати використання мережевих ресурсів;
- оптимізація витрат: допомагає ефективніше використовувати канали зв'язку та зменшувати витрати на інтернет-трафік.

Отже, глибоке розуміння сутності мережевого трафіку, його різновидів та особливостей стає ключовим етапом у процесі створення дієвої системи адміністрування корпоративної мережі.

1.3 Аналіз методів керування трафіком

Керування мережевим трафіком у компаніях – це непросте завдання, яке вимагає комплексного підходу. Його мета – забезпечити ефективну передачу даних, стабільну роботу мережевої інфраструктури та підтримувати бажаний рівень якості обслуговування. В основі цього процесу лежать загальноприйняті правила, що відображають найкращі методи раціонального використання обмежених мережевих ресурсів та досягнення високої продуктивності системи.

Перший ключовий принцип керування трафіком – це пріоритезація. Вона означає розподіл мережевих потоків залежно від їхньої важливості для користувачів або бізнес-завдань. Надаючи різний рівень доступу та пропускної здатності для різних типів трафіку, можна уникнути перевантаження мережі та забезпечити безперебійну роботу найважливіших сервісів, навіть коли мережеві ресурси обмежені. Наприклад, голосовий та відеозв'язок потребують мінімальних затримок і втрат пакетів, тому в системах керування їм зазвичай надається вищий пріоритет порівняно з такими неінтерактивними потоками, як передача великих файлів [5].

Інший важливий принцип – це адаптивність, тобто здатність системи керування реагувати на зміни в реальному часі. Оскільки мережевий трафік може значно коливатися залежно від часу доби, активності користувачів або появи нових сервісів, система керування повинна гнучко змінювати правила маршрутизації, контролю швидкості та обробки пакетів. Така гнучкість досягається завдяки використанню алгоритмів зворотного зв'язку, аналітики в реальному часі та методів прогнозування навантаження [5].

Також важливим є принцип справедливості, який забезпечує рівномірний розподіл доступу до ресурсів між усіма користувачами та сервісами. У корпоративному середовищі важливо не тільки надавати перевагу критично важливим додаткам, але й не допускати повного витіснення менш пріоритетного трафіку. Реалізація цього принципу базується на застосуванні алгоритмів керування чергами та політик обмеження швидкості [5].

Безпека є ще одним фундаментальним принципом, тісно пов'язаним з керуванням трафіком. Контроль над потоками даних дозволяє виявляти потенційно небезпечні дії, обмежувати або блокувати підозрілу активність, запобігати витоку інформації та впроваджувати політики доступу до мережевих ресурсів. Аналіз трафіку в цьому контексті слугує інструментом для виявлення атак, вторгнень або нетипової поведінки.

Економічна доцільність управління трафіком полягає в оптимізації витрат на інфраструктуру за рахунок більш раціонального використання існуючих каналів зв'язку, мінімізації потреби в надлишкових ресурсах та запобігання простою. Отже, управління трафіком виступає не лише як технічне рішення, а й як економічний інструмент для підвищення ефективності мережі.

1.4 Моделі керування трафіком

Управління трафіком у корпоративних мережах виступає не лише технічною задачею, але й предметом чисельних теоретичних досліджень, результатом яких стало формування широкого набору моделей, що відображають різні аспекти функціонування мережі. Моделі управління трафіком дозволяють формалізувати процеси передачі даних, прогнозувати навантаження, оцінювати ефективність алгоритмів керування та забезпечувати прийняття обґрунтованих рішень у реальному чи наближеному до реального часу форматі. Їх застосування є необхідним етапом як при проектуванні нових мережевих систем, так і при оптимізації існуючих інфраструктур.

У найзагальнішому вигляді всі моделі управління трафіком класифікують за низкою ознак: за рівнем абстракції, за підходом до опису потоків, за динамікою реагування та за характером розподілу ресурсів. Відповідно розрізняють математичні моделі з аналітичним описом параметрів мережі; імітаційні моделі, що моделюють роботу мережі на основі сценаріїв; статистичні моделі, побудовані на базі аналізу реальних даних; а також гібридні моделі, які поєднують елементи згаданих підходів.

Математичні моделі зазвичай ґрунтуються на інструментах теорії масового обслуговування, графів, теорії ймовірностей або диференціальних рівнянь. Вони дозволяють точно описати поведінку мережі за заданих умов, але часто потребують спрощень, що обмежує їхню придатність для складних динамічних систем. Наприклад, класичні моделі M/M/1 або M/M/n можуть застосовуватися для аналізу затримок і пропускної здатності під час обробки трафіку на маршрутизаторах або комутаторах [6].

Імітаційні моделі надають змогу відтворювати реальний трафік у віртуальному середовищі за допомогою спеціалізованого програмного забезпечення, зокрема NS-3, OPNET або Cisco Packet Tracer. Вони дозволяють проводити експерименти з різними налаштуваннями мережі, вивчати поведінку при змінах навантаження, впроваджувати нові політики управління та аналізувати їх вплив на ефективність системи. Основною перевагою таких моделей є висока гнучкість та близькість до реальних умов експлуатації [6].

Статистичні моделі спираються на аналіз великих обсягів історичних даних про мережеву активність. Вони дозволяють будувати профілі поведінки трафіку, виявляти закономірності, аномалії, циклічні зміни та інші тренди. Такі

моделі ефективні для прогнозування навантаження та в системах виявлення вторгнень або аномальної активності [6].

Іншою важливою категорією є моделі на основі принципів оптимізації, зокрема лінійного чи нелінійного програмування, які дозволяють визначати найкращі варіанти розподілу мережевих ресурсів за заданими обмеженнями. Такі моделі особливо актуальні, коли ставиться задача мінімізувати затримку або підвищити пропускну здатність мережі при одночасному обслуговуванні великої кількості користувачів або сервісів.

Окремо слід згадати моделі, що реалізують концепцію самонавчання або адаптації. До них належать системи з використанням нейронних мереж, алгоритмів машинного навчання або методів нечіткої логіки. Їхньою особливістю є здатність до самостійного вдосконалення в процесі експлуатації, вивчення нових патернів трафіку та автоматичне коригування політик керування без втручання адміністратора.

Вибір конкретної моделі управління трафіком залежить від завдань мережі, її масштабу, складності, доступних технічних засобів і кваліфікації персоналу. У багатьох випадках ефективним є застосування гібридних підходів, що поєднують точність аналітичних методів із гнучкістю імітаційного або адаптивного моделювання. Такий підхід дозволяє максимально точно відтворювати поведінку мережі та приймати ефективні управлінські рішення в умовах постійної зміни інформаційного середовища.

Усі зазначені принципи функціонують у тісному взаємозв’язку й утворюють концептуальну базу для розробки систем управління трафіком у сучасних корпоративних мережах. Їхнє дотримання забезпечує баланс між якістю обслуговування та ефективністю.

На рис. 1.3 подано таблицю з порівняльним аналізом програмного забезпечення для імітаційного моделювання мережевого трафіку. Згідно з проведеним аналізом спеціалізованих інструментів для імітаційного моделювання трафіку можна сформувати низку узагальнених висновків, що мають як теоретичне, так і практичне значення. Кожен із розглянутих інструментів демонструє свої унікальні переваги й обмеження, які визначають доцільність використання залежно від поставлених завдань, рівня підготовки користувача та контексту застосування — навчального, дослідницького чи практичного.

Назва ПЗ	Тип моделювання	Платформа	Рівень складності
Cisco Packet Tracer	Візуальна імітація мережі	Windows, Linux	Низький
GNS3	Емуляція мережі з реальним трафіком	Windows, Linux, macOS	Середній
NS-3	Імітаційне моделювання подій	Windows, Linux	Високий
OPNET (Riverbed Modeler)	Імітаційне моделювання	Windows	Високий
OMNeT++	Імітаційне моделювання подій	Windows, Linux	Високий
Назва ПЗ	**Переваги**	**Недоліки**	
Cisco Packet Tracer	Легкість у використанні, інтерактивність	Обмеженість у складних сценаріях	
GNS3	Реалістична емулювання, підтримка реального ПЗ	Потребує потужних ресурсів	
NS-3	Висока точність, підтримка скриптів	Складність у використанні	
OPNET (Riverbed Modeler)	Масштабованість, точне моделювання	Комерційне ПЗ, складність налаштування	
OMNeT++	Гнучкість, модульна структура	Складність для початківців	

Рисунок 1.3 – Порівняння ПЗ, що має функцію імітації трафіку

Cisco Packet Tracer вирізняється простотою використання та зручним інтерфейсом, завдяки чому є популярним серед студентів і початківців. Проте його функціональні можливості суттєво обмежені, особливо в контексті відтворення реалістичного мережевого середовища або роботи з трафіком у динамічних умовах. На протилежний бік стоїть GNS3, який забезпечує високу реалістичність завдяки можливості емулювати справжнє мережеве обладнання та взаємодії з реальними операційними системами, що робить його ефективним для тестування конфігурацій у середовищі, наближеному до реального. Проте для коректної роботи GNS3 потребує значних апаратних ресурсів і базових навичок адміністрування. [7]

NS-3 та OMNeT++ належать до засобів високої складності, орієнтованих переважно на дослідницькі цілі та моделювання складних систем передачі даних. Їх використання передбачає глибоке розуміння мережевих архітектур, знання мов програмування та здатність працювати з математичними моделями процесів. Особливістю NS-3 є висока точність і гнучкість, що дозволяє створювати складні сценарії трафіку на основі реальних статистичних даних. OMNeT++ приваблює своїм модульним принципом, що полегшує масштабування моделей та розширення їх новими функціональними блоками. Також значущим інструментом є OPNET (тепер Riverbed Modeler), який забезпечує деталізоване моделювання на різних рівнях мережевої архітектури і є корисним у складних корпоративних або операторських інфраструктурах. Недоліком є складність освоєння та комерційна ліцензійна модель, що обмежує доступність для широкого кола користувачів [7].

Загалом можна констатувати, що жоден із розглянутих інструментів не є універсальним. Вибір програмного забезпечення повинен залежати від цілей дослідження або навчання, характеру мережевого середовища, рівня технічної підготовки фахівця та доступних ресурсів. У багатьох випадках оптимальним підходом є комбінування кількох засобів для досягнення найкращого балансу між простотою, точністю та функціональністю.

2 АНАЛІЗ МЕТОДІВ КЕРУВАННЯ ТРАФІКОМ

2.1 Класифікація методів керування трафіком

У найзагальнішому формулюванні класифікація методів управління трафіком ґрунтується на кількох ключових критеріях.

По-перше, за способом збору та обробки даних про стан мережі розрізняють статичні та динамічні методи. Статичні методи передбачають фіксовані правила й параметри, встановлені адміністратором мережі до початку передачі даних, що залишаються незмінними незалежно від змін навантаження або топології. Вони характеризуються простою реалізацією, але обмежені в здатності реагувати на непередбачені ситуації або пікові навантаження. Натомість динамічні методи реалізують адаптивне управління на основі поточних показників трафіку, забезпечуючи гнучку реакцію на зміни умов у реальному часі [8].

По-друге, за характером впливу на трафік методи поділяють на превентивні, регулятивні та реактивні. Превентивні підходи спрямовані на попередження перевантажень шляхом встановлення обмежень на рівні політик доступу чи фільтрації даних. Регулятивні методи забезпечують постійний моніторинг та коригування параметрів мережевого середовища, зокрема шляхом управління чергами або перерозподілу смуги пропускання. Реактивні методи активуються у відповідь на конкретні події або відхилення у роботі мережі, зокрема при виявленні перевантаження, втрат пакетів або підозрілої активності [8].

Іншим важливим критерієм є рівень застосування у моделі мережі. Існують методи управління трафіком, реалізовані на фізичному, канальному, мережевому, транспортному або прикладному рівні. Так, обмеження пропускної здатності та фільтрація MAC-адрес належать до методів нижчих рівнів, тоді як розподіл ресурсів між застосунками або контроль за сесіями користувачів - до прикладного рівня. Ефективне управління трафіком вимагає всебічного підходу з урахуванням усіх рівнів мережевої моделі, що особливо актуально для великих корпоративних структур із багаторівневою архітектурою.

Окрему категорію становлять інтелектуальні методи, які застосовують алгоритми машинного навчання, нейронні мережі або інші засоби штучного інтелекту для прийняття рішень щодо маршрутизації, пріоритезації чи виявлення аномалій. Їхня класифікація може ґрунтуватися на типі алгоритмів, рівні автономності системи, етапі навчання (з учителем або без нього) тощо. Подібні методи демонструють високу ефективність у складних, динамічних та гетерогенних мережах, однак потребують значного обсягу даних для навчання та значних обчислювальних ресурсів [8].

Узагальнюючи, класифікація методів управління трафіком дозволяє систематизувати підходи до організації контролю за потоками даних, полегшує вибір оптимального інструменту для конкретної мережевої ситуації та формує основу для подальшого вдосконалення існуючих або розробки нових механізмів керування в умовах зростаючої складності сучасних інформаційних систем. Класифікацію засобів моніторингу можна подати у вигляді кількох груп, зображених на рис. 2.1 [8].

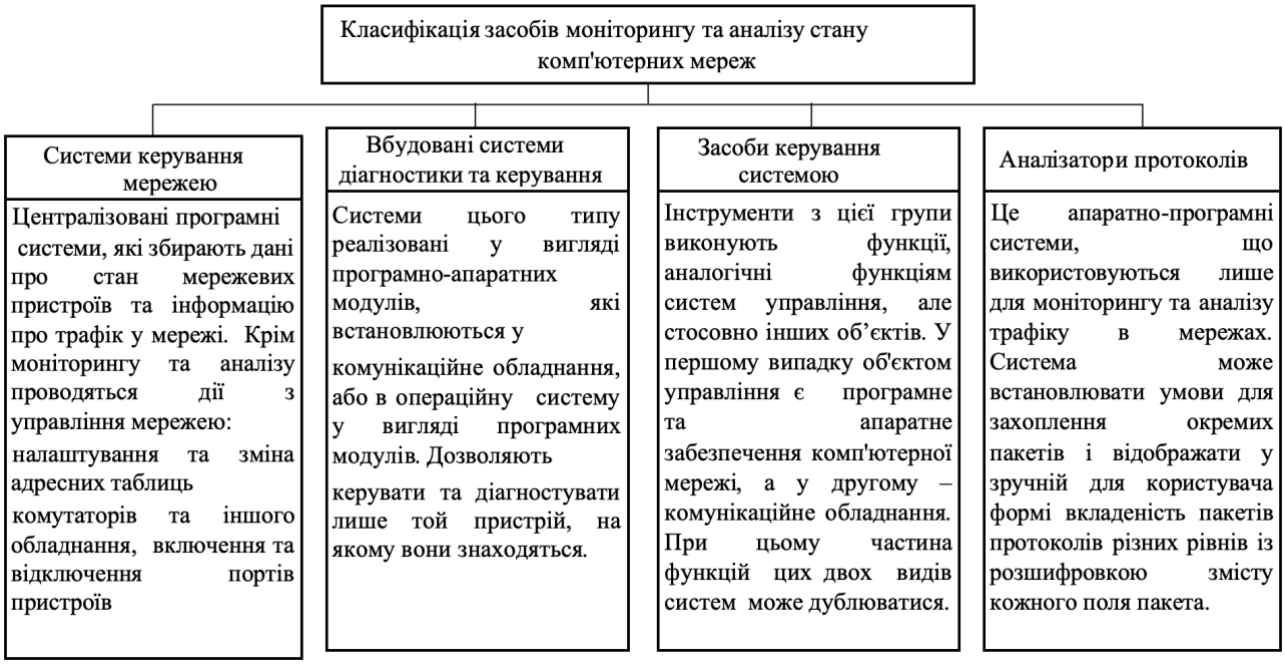

Рисунок 2.1 - Класифікація засобів моніторингу та аналізу стану комп'ютерних мереж

Система керування інформаційним потоком являє собою сукупність взаємопов’язаних компонентів, які можуть змінювати стан, параметри та

характеристики потоку. Деякі методи регулювання інформаційного потоку можуть бути репрезентовані у вигляді груп, як показано на рис. 2.2 [8].

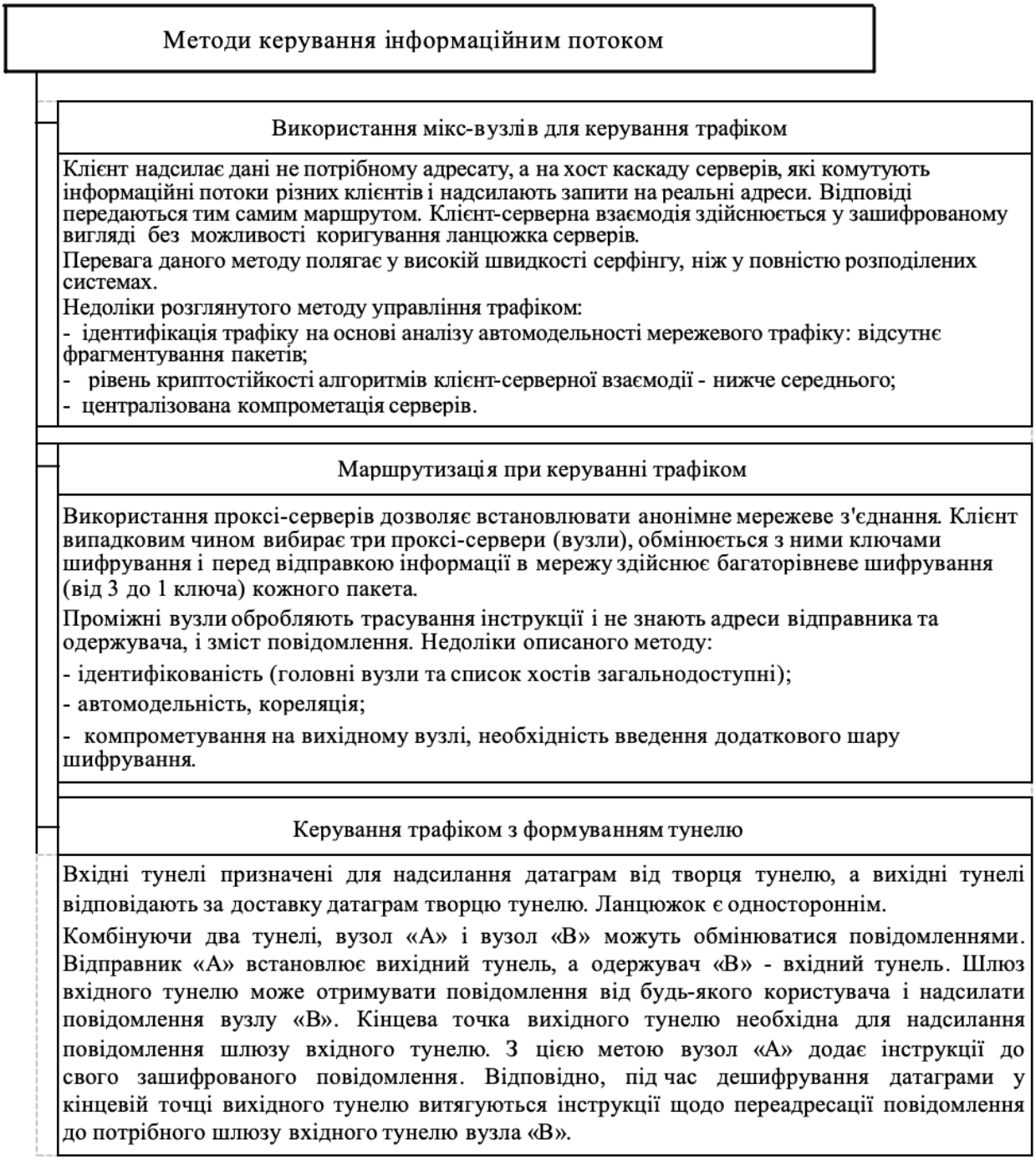

Рисунок 2.2 - Методи управління інформаційним потоком

До завдань підсистеми регулювання інформаційного потоку належать:

- забезпечення користувача трафіком з якістю, що максимально його задовольнить;

- ефективне управління ресурсами комп'ютерної мережі на високому рівні;
- у мінімально можливі терміни застосування заходів з мінімізації тривалості стану перевантаження мережі [9].

Загальна схема системи управління інформаційними потоками показана на рис. 2.3.

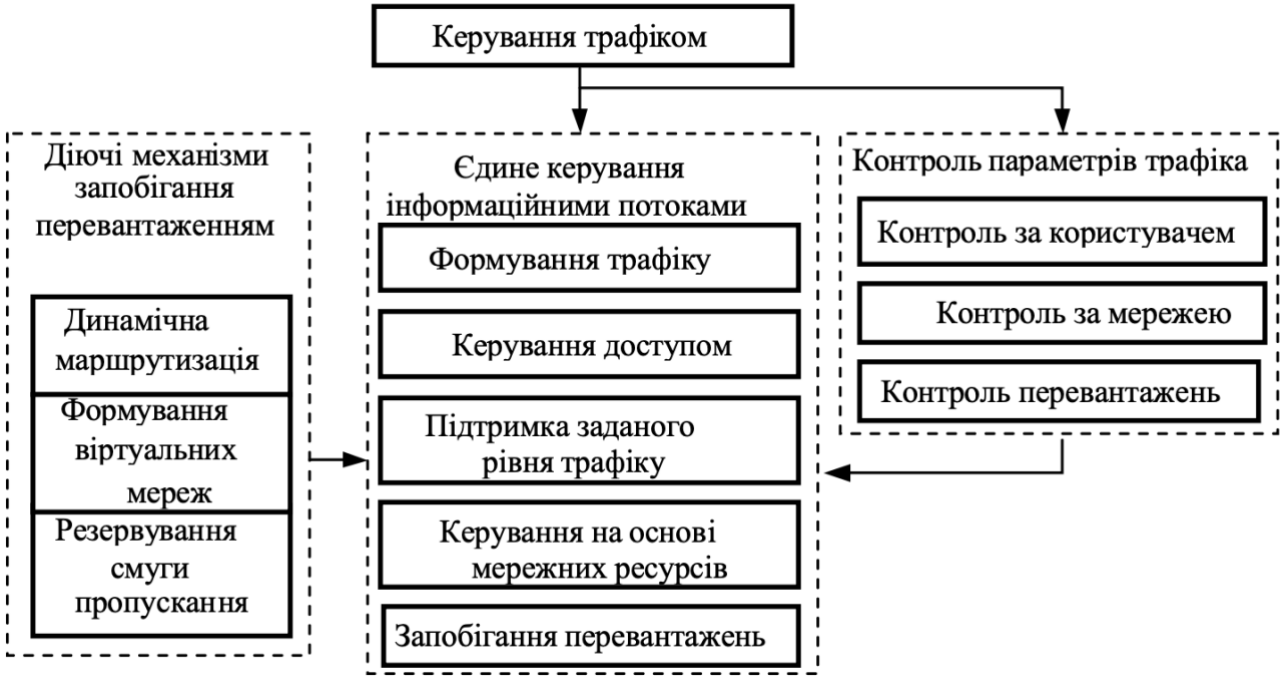

Рисунок 2.3 - Загальна схема системи управління інформаційними потоками

Підсистема управління пропускною здатністю в комп'ютерних мережах (КМ) визначається як сукупність норм та операцій, спрямованих на організацію прийому й передачі пакетів за рахунок мережевого інтерфейсу. Вона застосовує алгоритми, які встановлюють прийнятні швидкості та формати пакетів, що надходять на вхідний інтерфейс, а також характеристики швидкості, які мають бути забезпечені на вихідному інтерфейсі [9].

Управління пропускною здатністю може включати алгоритми утримання пакетів, зміну черговості відправлення та пріоритетності пакетів. Черговість передачі може бути композицією кількох підчерг. Також зауважимо, що підсистема управління трафіком являє собою потік, що з'єднує два хости.

Із шести механізмів управління пропускною здатністю, які застосовуються у КМ, виокремлюють:

- обмеження вихідного трафіку (shaping);
- обмеження вхідного трафіку (policing);
- знищення пакетів (dropping);
- планування (scheduling);
- класифікацію (classification);
- маркування (marking).

Запропонований метод організації підсистеми управління трафіком у каналі передачі даних базується на класах даних. Відповідно визначено завдання розробки методу ідентифікації класу трафіку та механізму його управління.

На рис. 2.4 продемонстровано структурну схему модуля управління пропускною здатністю, що складається з двох підмодулів - модуля ідентифікації трафіку та модуля класифікації й обмеження трафіку, де відображено траєкторію руху трафіку.

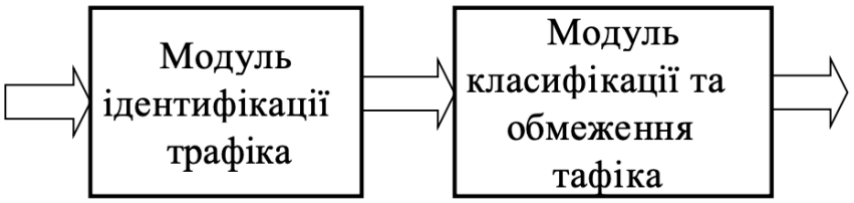

Рисунок 2.4 - Модуль управління пропускною здатністю

У модулі ідентифікації кожен тип даних набуває власного ідентифікатора. Після цього він надходить до модуля управління трафіком, де за допомогою попередньо отриманої мітки здійснюється обробка відповідним класом. Робота модуля управління пропускною здатністю ілюструється на рис. 2.5.

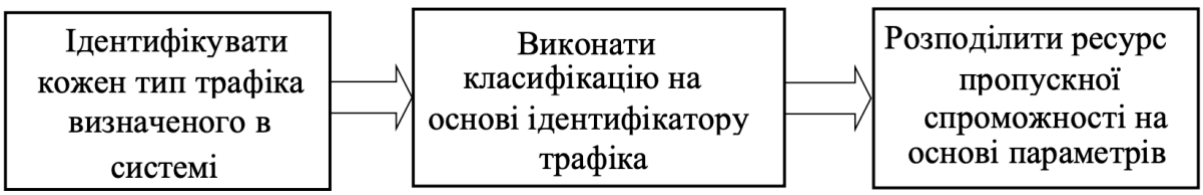

Рисунок 2.5 - Структурна схема підсистеми управління пропускною здатністю

В представленому методі класифікації передбачено повне розпізнавання всього інформаційного потоку. Після присвоєння їм ідентифікатора вони переходять до класу, де визначаються пріоритет, швидкість передачі та інші параметри. Також реалізуються механізми розподілу пропускної здатності каналу зв’язку та перенаправлення трафіку в комп’ютерній мережі. Структурна схема вирішення задачі методу управління трафіком у системі КМ зображена на рис. 2.6.

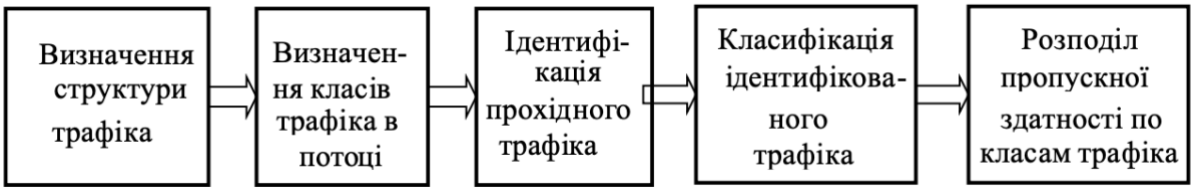

Рисунок 2.6 - Функціональна схема завдання щодо організації системи управління трафіком

Одним із етапів розробки системи формування трафіку є ідентифікація виду даних, що проходять через неї (з урахуванням запропонованої класифікації за типами). Ідентифікація поточного трафіку показана на рис. 2.7.

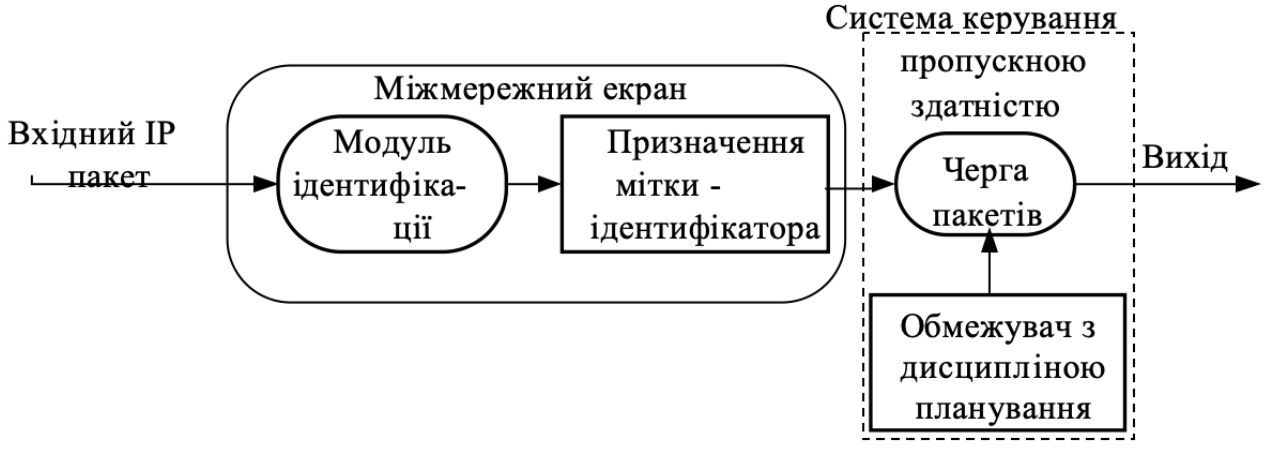

Рисунок 2.7 - Структурна схема ідентифікації потоку, що проходить

Управління трафіком у комп'ютерній мережі забезпечує:

- уникнення потреби ручного налаштування мережевих пристроїв для визначення конкретних маршрутів;

- оцінку пропускної здатності каналу та обсягу трафіку під час прокладання маршруту через опорну мережу;

- у разі відмови вузла - корекцію топології опорної мережі за рахунок механізмів динамічної адаптації.

2.2 Сучасні методи балансування навантаженням

Оптимізація використання розподілених ресурсів у комп'ютерних мережах є одним із ключових напрямів сучасних досліджень та практичної діяльності. Ефективність розподіленого середовища безпосередньо залежить від успішності розв'язання цього завдання, що відображається у підвищенні продуктивності обчислювальних вузлів, масштабованості системних модулів та оптимізації мережевого трафіку [10].

Значна частина наявних методів інтеграції ресурсів у розподілених системах базується на класичних підходах, які мають беззаперечні переваги, є детально вивченими та широко впровадженими. До них належать методи математичної статистики, лінійного програмування, теорії ймовірностей, теорії графів тощо. Висока адекватність моделей, у яких функціонують ці методи, гарантується потужністю математичного апарату та ґрунтовністю відповідних теоретичних засад.

Водночас потенціал для оптимізації внутрішньосистемної взаємодії у розподілених середовищах залишається значним. Застосування інноваційних підходів, що суттєво відрізняються від традиційних методологій, дозволить підвищити ефективність поточних моделей управління розподіленими ресурсами.

Нижче наведено базову класифікацію стратегій балансування навантаження. У цьому та наступних розділах аналізуються виключно автоматизовані стратегії, реалізовані на рівні системного або проміжного програмного забезпечення (middleware), інтегровані в інструментальні комплекси або безпосередньо в прикладні розподілені системи. Ручні та напівавтоматичні методи балансування навантаження не розглядаються як перспективні, оскільки вони належать до менш ефективних інструментів формування стратегій розподілу ресурсів.

За рядом істотних ознак можна умовно виділити такі основні класи стратегій [10]:

1. Відповідно до принципу врахування динаміки, стратегії розподілу навантаження класифікують на статичні, напівдинамічні та динамічні. Статична стратегія передбачає використання фіксованого плану розподілу, визначеного заздалегідь. Напівдинамічна стратегія базується на формуванні плану на етапі ініціалізації, що передує безпосередньому виконанню обчислень; попередній аналіз ресурсів у цьому контексті дозволяє підвищити ефективність порівняно зі статичним підходом. Динамічна стратегія застосовується, коли план розподілу коригується в процесі виконання завдань - згідно з графіком або під впливом зовнішніх факторів - шляхом перерозподілу обчислювальних операцій між вузлами мережі відповідно до актуального на поточний момент оптимального плану. Динамічне балансування є складним інструментом: незважаючи на потенціал для значного підвищення продуктивності, його неефективне впровадження може призвести до зниження корисного навантаження [10].

На відміну від статичних та напівдинамічних методів, динамічні засоби балансування навантаження адаптовані до мінливих умов експлуатації. Їхня ефективність є найбільш вираженою у системах, де параметри обчислювальних процесів не визначені заздалегідь через вплив внутрішніх або зовнішніх чинників, таких як специфіка міжпроцесної взаємодії чи динаміка розподілу ресурсів. Варто зазначити, що статичні стратегії залишаються актуальними завдяки своїй простоті та продуктивності у випадках, коли обчислювальні завдання є прогнозованими, а апріорна інформація про систему — доступною. Проте, з огляду на існуючі обмеження, їх застосування в динамічних середовищах не завжди є доцільним. Ключовими аспектами динамічних стратегій є механізми міграції ресурсів, зокрема процесів та даних. При реалізації перенесення навантаження в гетерогенних середовищах виникає необхідність вирішення критично важливих завдань, пов'язаних із забезпеченням сумісності, портируваності, інтероперабельності та масштабованості архітектури.

2. За ступенем адаптивності до зміни навантаження стратегії розподілу поділяються на адаптивні та неадаптивні. Адаптивні стратегії забезпечують балансування навантаження та перерозподіл ресурсів за умов зміни конфігурації розподіленої системи. Зокрема, у разі додавання нових вузлів або виходу з ладу наявних здійснюється автоматичне перепланування обчислень для підтримки ефективності системи. Варто зазначити, що поняття «адаптивне

балансування навантаження» часто ототожнюється з динамічним балансуванням у вузькому значенні [10].

3. За методологією зміни плану розподілу стратегії класифікуються на залежні та незалежні. Залежний розподіл передбачає моніторинг подій, пов'язаних зі зміною характеристик навантаження, або таймерних подій, із подальшим формуванням актуального плану. Незалежний розподіл (що охоплює напівдинамічні та динамічні типи) базується на апріорному обчисленні послідовності планів до початку виконання розподіленого додатка. Під час виконання відбувається зміна статичних планів згідно з заздалегідь заготовленим сценарієм, прив’язаним до певних подій системи або цільового обчислювального процесу, таймерних відліків - без повторного обчислення самих планів: $П_1 \rightarrow П_2 \rightarrow \ldots \rightarrow П_k$. Таке змінювання може здійснюватися у точках бар’єрної синхронізації паралельно-розподіленого додатка. Цей підхід також може називатися квазидинамічним [10].

4. За принципом управління та характером відповідальності за розподіл навантаження алгоритми балансування поділяються на централізовані та децентралізовані. Вибір стратегії визначає механізм розподілу ресурсів у системі. Централізовані алгоритми класифікуються як глобальні, оскільки передбачають наявність центрального вузла (планувальника), що збирає дані з усіх елементів системи для прийняття управлінських рішень. Наявність такого диспетчера та потреба в постійній синхронізації інформації про стан ресурсів часто стають «вузьким місцем» та критичною вразливістю подібних стратегій. Децентралізовані алгоритми класифікуються як локальні, оскільки не потребують агрегації даних про стан усіх вузлів розподіленої системи. Процес розподілу ресурсів у таких схемах здійснюється кожним вузлом автономно, іноді з урахуванням взаємодії з найближчими сусідами. Децентралізовані підходи є оптимальними для мережевих обчислень та сценаріїв, де топологія комунікацій між вузлами адаптована до специфіки конкретних прикладних завдань. Централізовані стратегії балансування використовуються найчастіше через універсальність підходу та простоту алгоритмізації, хоча й не завжди забезпечують хорошу масштабованість, проте, як уже було зазначено, для деяких прикладних задач децентралізований підхід виявляється ефективнішим, незважаючи на його обмежену застосовність та певні складнощі при спільній синхронізації та забезпеченні цілісності всього обчислювального процесу [10].

5. За критерієм універсальності стратегії класифікують на універсальні та спеціалізовані. До останніх належать стратегії, адаптовані до архітектур розподілених систем, конкретних топологій мережевого середовища, специфічних алгоритмів або інфраструктурних особливостей. Своєю чергою, універсальні стратегії призначені для підтримки широкого спектра алгоритмів, є інваріантними до сфери застосування, а також вирізняються легкістю інтеграції в додатки та уніфікованістю інтерфейсів [10].

6. З огляду на можливість передбачення динамічних змін, стратегії поділяють на прогнозні та такі, що не володіють здатністю до передбачення майбутніх станів (зокрема, коливань навантаження). Очевидно, що архітектури, оснащені ефективними інструментами короткострокового та довгострокового прогнозування обчислювальних процесів, демонструють значні переваги в адаптивності порівняно зі стандартними системами планування обчислень. Ключовою метою при розробці прогностичних стратегій є підвищення точності прогнозування, оскільки цей показник прямо корелює з ефективністю мінімізації непродуктивних втрат. Слід зазначити, що використання надмірно складних та ресурсомістких алгоритмів, попри їхню високу точність, не завжди забезпечує приріст продуктивності. Навпаки, надмірне споживання обчислювальних ресурсів може негативно вплинути на загальну ефективність системи. Водночас низька точність прогнозів також є фактором, що знижує результативність функціонування системи [10].

У межах дослідження доповнено наведену вище класифікацію шляхом типізації стратегій балансування навантаження за низкою релевантних додаткових ознак. До розгляду не було включено параметри, які не мають істотного значення для більшості сфер застосування (зокрема, класифікації за ступенем гранулярності ресурсів, пріоритезацією обчислень, прозорістю тощо).

7. Стратегії розподілу навантаження можна класифікувати залежно від того, чи враховують вони фактори, що зумовлюють розбалансування системи. До таких факторів належать як внутрішні (пов'язані з функціонуванням розподіленого додатка), так і зовнішні (незалежні від нього) чинники. Врахування цих аспектів суттєво підвищує прогностичну спроможність стратегій.

8. За критерієм оцінюваних характеристик стратегії розподілу поділяються на ті, що враховують виключно продуктивність обчислювальних вузлів системи, та ті, що додатково беруть до уваги пропускну здатність

мережевої інфраструктури. У межах останнього підходу передбачається детальний аналіз параметрів інформаційного та керуючого трафіку.

9. За ступенем точності оцінювання доступних ресурсів стратегії розмежовують на наближені методи та методи, що базуються на реалістичних моделях із використанням комплексних агрегатних показників споживання ресурсів (наприклад, такими, що включають такі часткові характеристики, як кількість активних потоків виконання у вузлі, кількість і швидкість процесорів вузла, обсяг доступної оперативної та віртуальної пам'яті вузла, міжплатформні накладні витрати тощо).

10. За ініціатором процесу розподілу алгоритми балансування навантаження поділяються на ті, де ініціатором виступає приймач (недовантажені вузли), та ті, в яких ініціатором є джерело навантаження (перевантажені вузли).

2.3 Прогностичні стратегії балансування навантаження

Згідно з наведеною класифікацією, ключове значення для великомасштабних гетерогенних розподілених мереж мають динамічні адаптивні стратегії, здатні прогнозувати зміни навантаження на рівні окремих вузлів, сегментів мережі та системи загалом (рис. 2.8).

Рисунок 2.8 - Класи стратегій балансування навантаження

Попри те, що прогностичні моделі в контексті управління ресурсами тривалий час розглядалися переважно на теоретичному рівні, їх впровадження демонструє значний потенціал для оптимізації використання невикористаних ресурсів та є дієвим інструментом підвищення загальної продуктивності розподілених мережевих систем. Широкий доступ до ідей через розподілені, різноманітні обробки, що підтримуються прогресами в мережевих технологіях, дає нам змогу з упевненістю зробити висновок, що дослідження балансу навантаження через передбачення є дуже актуальними. Довготермінові прогнози (мікропрогнозування) і короткострокові (мекропрогнози) використовуються для визначення довгострокових наслідків цих цілей. Перший тип прогнозування корисний, коли ви плануєте взяти досить великий проміжок часу і потребує важливої оцінки загальних ресурсів з довгостроковими факторами. Мікропрогноз зазвичай виконується у певному темпі, вимірюваному, і короткострокові цілі виконання можна розглядати як нагороду від прогнозу, що базується на обчисленій вартості створення вашого власного прогнозу [11].

Стратегію прогнозу можна розділити на два типи: централізовану і децентралізовану. У першому випадку, прогнозом є єдина активна точка, створена за допомогою даних, зібраних з кожного вузла у мережі. У другому випадку кожен вузол виконує власний незалежний прогноз щодо власної зміни навантаження, отже існує багато активних точок. Зауважте, що централізоване і децентралізоване передбачення не обов'язково має відповідати централізованому та децентралізованому керуванню. Список стратегій, які були реалізовані до того, тому що результати прогнозів з різних комп'ютерів можуть бути використані центральним агентом і тд.

Щоб побачити проблему з обговоренням цього прикладу, уявіть собі систему, яка розподілена і має навантаження, яке постійно змінює декілька обчислювальних процесів одного розділеного додатка. Для спрощення ми припускаємо, що кожен процес виконується окремо від вузла. Процеси виконуються у фоні (за допомогою програми можна налаштувати виконання операційних процесів та визначених користувачем процесів на вузлі). Процеси узгоджено синхронізуються в визначених точках, і кожен із них під час синхронізаційного кроку виконує обсяг розрахунків, пропорційний продуктивності свого вузла. Якщо розподілити навантаження виходячи з

часових інтервалів, то всі процеси мають виконати свої частки обчислень приблизно за однаковий проміжок часу (рис. 2.9).

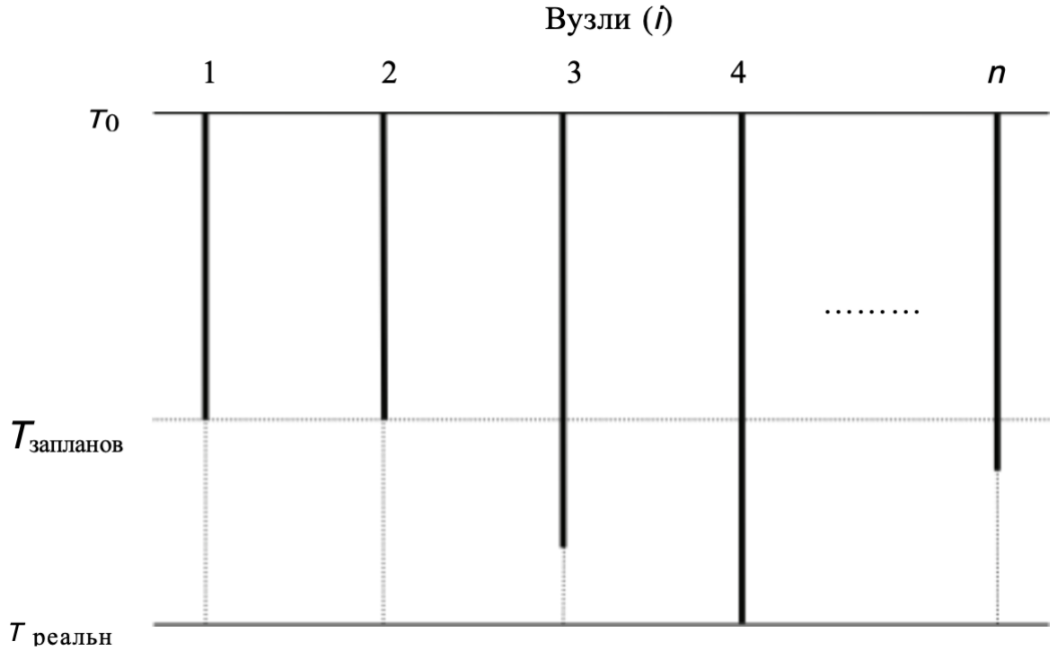

Рисунок 2.9 – Розподіл навантаження по вузлам мережі

Однак у вузлах можуть виникати події, що уповільнюють локальний фоновий процес. Це зазвичай пов’язано з активністю основних процесів. У результаті цільові завдання завершуються пізніше запланованого часу, що може призвести до ситуації, коли «семеро одного чекають». Тобто процеси, які затягнулись, спричиняють непродуктивний простій (втрату ресурсів), масштаби якого залежать від кількості вузлів, які завершили свою роботу вчасно:

$$M = \sum_{i=1}^{n} \left(T_{реальн}(i) - T_{запланов}(i) \right), \tag{2.1}$$

Завдання прогнозування полягає у мінімізації функції втрат M шляхом визначення меж $T_{реальн}(i)$ і, відповідно, перерозподілу обчислювальних часток (навантаження) з урахуванням розрахованого прогнозу та подальшого перерозподілу обчислювальних ресурсів на основі отриманих прогнозних даних.

Для динамічної адаптації до постійних змін продуктивності вузлів розподіленої системи впроваджується прогностична модель розподілу

обчислювального навантаження. Зміна навантаження вузла представляється у вигляді послідовності вибіркових значень, отриманих у результаті періодичних вимірювань. Інтегральною характеристикою системи є сукупність таких послідовностей для всіх вузлів мережі.

Класичний метод аналізу базується на застосуванні статистичних інструментів для обробки часових рядів. Процес зміни навантаження, представлений як дискретна множина значень завантаженості (або вільних ресурсів), виміряних через визначені інтервали часу, піддається статистичній обробці. З огляду на природу інформаційних процесів, на певних часових масштабах у структурі даних можна виокремити детерміновану складову (періодичну та неперіодичну), яка за критерієм «сигнал/шум» (SNR) суттєво відрізняється від випадкової компоненти. Виявлені закономірності дозволяють здійснювати ймовірнісне прогнозування майбутнього завантаження вузлів, зокрема методами екстраполяції, для оптимізації розподілу ресурсів [11].

При підготовці часових рядів ключове значення має вибір інтервалу дискретизації. Занадто малий крок призводить до зростання непродуктивних витрат, тоді як надмірно великий інтервал спричиняє ризик збільшення простоїв у разі похибки прогнозу. Вибір кроку також залежить від рівня гранулярності розподілу обчислювальних ресурсів.

Допускається масштабування значень ряду шляхом накопичувальної децимації, що передбачає об’єднання декількох часових інтервалів в один із відповідним підсумовуванням навантаження. У низці випадків доцільно використовувати неоднорідну дискретизацію зі змінним кроком, зокрема нелінійного характеру, що дозволяє отримати деталізовану інформацію про динаміку навантаження у критичних сегментах системи. Замість абсолютних показників можуть використовуватися нормовані значення [0, … , 1], що відображають коефіцієнт використання ресурсів. Крім того, аналіз може базуватися на розрахунку різниці між максимальними та мінімальними показниками навантаження в межах обраного масштабного періоду (рис. 2.10).

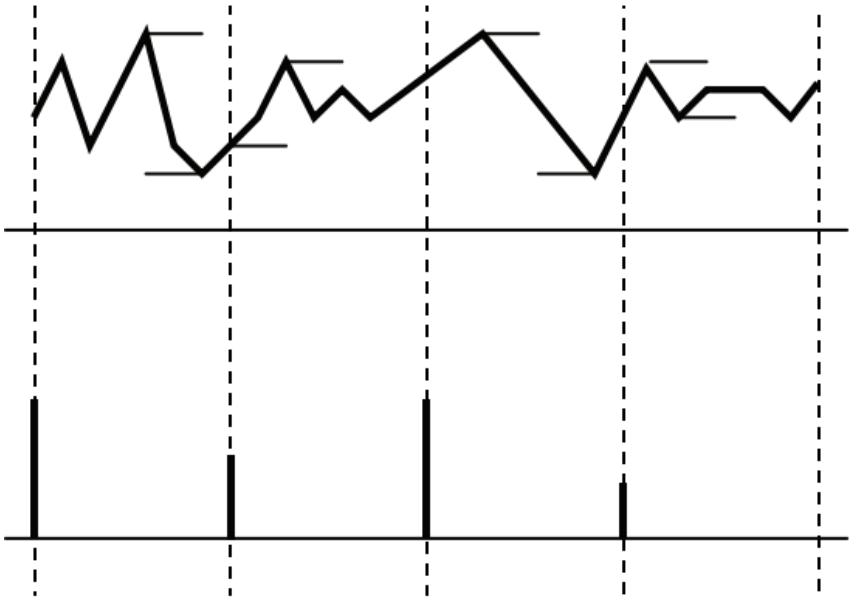

Рисунок 2.10 - Формування ряду на основі диференційних значень

Для побудови інтегрованої моделі, що відображає динаміку функціонування розподіленої системи, доцільно застосовувати багатовимірний аналіз часових рядів, сформованих на основі паралельних та незалежних вимірювань у всіх вузлах мережі. Для ідентифікації закономірностей вищого порядку доцільно залучати інструментарій математичної статистики, зокрема методи обчислення кроскореляції.

Варто зауважити, що попри детерміновану програмну природу таких систем, прогнозування їхньої продуктивності з високим ступенем точності є ускладненим. Вказані системи за своїми характеристиками належать до складних динамічних систем, що зумовлює необхідність застосування синергетичного підходу в межах теорії динамічних систем для їхнього подальшого дослідження.

2.4 Прогнозування на основі фрактального методу

Альтернативний підхід до розроблення стратегії прогнозування базується на застосуванні фрактальних принципів та синергетичної парадигми інформаційних процесів у складних системах. Гіпотеза щодо нелінійної динаміки робочого навантаження ґрунтується на фундаментальних положеннях теорії детермінованого хаосу та динамічних систем. Передбачається, що розподілене середовище функціонує як нелінійна динамічна система, поведінка якої характеризується хаотичним квазівипадковим рухом із певним рівнем

аутопоетичної організації. Це проявляється у виникненні стохастичних флуктуацій в обчислювальному середовищі під впливом дестабілізуючих чинників. Відповідно до визначення Ч.Л. Сайерса, процес вважається детерміновано-хаотичним, якщо він генерується повністю детермінованою системою, що проявляється як результат безладно функціонуючих послідовностей у межах стандартних часових інтервалів [11].

З огляду на обмежену ефективність класичних лінійних моделей при описі складних нелінійних процесів, пропонується використання фрактальної моделі [11], яка є більш адекватною до реальних умов функціонування систем. Фрактальний аналіз динаміки навантаження полягає у виявленні в часових траєкторіях структур із властивістю масштабної інваріантності. Одночасно доцільно проаналізувати зміну параметрів розподіленого середовища шляхом моделювання динаміки його розвитку за допомогою імовірнісних траєкторій у фазовому просторі станів, а також дослідити поведінку системи поблизу атракторів, які безпосередньо впливають на її продуктивність. Виявлена фрактальна самоподібність імовірнісних траєкторій може бути ефективно використана для прогнозування динаміки навантаження та оптимізації параметрів балансування інформаційного трафіку у віртуальному розподіленому середовищі. Для опису фрактальних структур доцільно застосовувати метод систем ітераційних функцій (IFS). Процедура ідентифікації фрактальних варіантів за своїми принципами є спорідненою з алгоритмами фрактального стиснення даних.

На рисунку 2.11 наведено приклад, у якому інваріант масштабування підтверджує наявність фрактальних ознак у часовому ряді, що дозволяє використовувати встановлені закономірності для прогнозування подальшого розвитку обчислювального процесу.

Підсумовуючи викладене, варто зазначити, що доцільною є інтеграція класичного та фрактально-синергетичного підходів із чітким розмежуванням сфер їхнього застосування та ефективності, а також розробка єдиної комплексної стратегії моделювання динаміки навантаження. У ситуаціях, коли витрати на ідентифікацію фрактальних характеристик динаміки виходять за межі економічної доцільності або не забезпечують необхідної точності, слід застосовувати інструментарій традиційних статистичних методів.

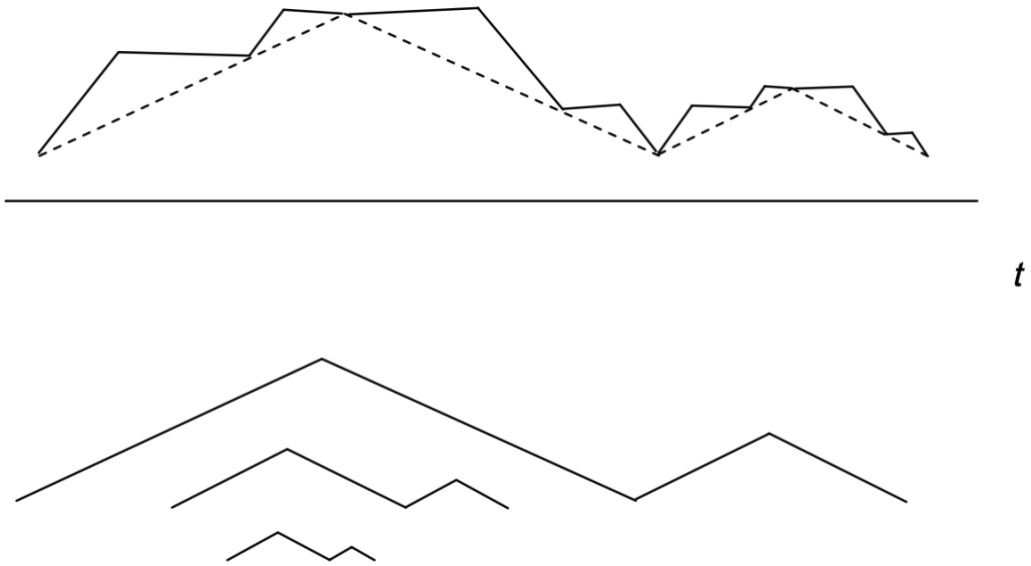

Рисунок 2.11 - Властивість фрактальної самоподібності ряду вимірів

3 АЛГОРИТМ ДИНАМІЧНОГО БАЛАНСУВАННЯ НАВАНТАЖЕННЯ

Сучасні корпоративні комп’ютерні мережі є складними інфраструктурними комплексами, що складаються з різноманітного мережевого обладнання (комутаторів доступу та розподілу, маршрутизаторів, міжмережевих екранів) та серверних ресурсів (веб-, поштових серверів, систем баз даних, документообігу та білінгу). Для забезпечення належного рівня відмовостійкості та безперервного контролю функціонального стану мережі впроваджуються системи моніторингу, які за принципом дії поділяються на активні та пасивні. Активні системи ініціюють запити до об’єктів моніторингу та обробляють отримані відповіді, тоді як пасивні базуються на безперервному аналізі мережевого трафіку.

Активні системи моніторингу корпоративних мереж класифікуються на централізовані та розподілені. Централізовані рішення передбачають наявність єдиного вузла для збору та агрегації даних. Розподілені системи використовують мережу вузлів, що оптимізує навантаження на систему моніторингу. Такі вузли можуть виконувати функції опитування об’єктів або виступати в ролі розподілених сховищ даних. В архітектурах такого типу навантаження, як правило, розподіляється статично [12].

У цій роботі розглядається концепція побудови розподіленої системи моніторингу, що використовує вільні обчислювальні ресурси серверів корпоративної мережі. Впровадження механізмів автоматичного налаштування агентів та динамічного балансування навантаження дозволяє суттєво скоротити час розгортання системи та знизити експлуатаційні витрати. При проектуванні топології системи та стратегії опитування необхідно враховувати накладні витрати на мережевий трафік, що генерується процесом моніторингу. У зв’язку з цим пропонуються алгоритми динамічного розподілу навантаження шляхом адаптивного призначення джерел даних.

Серед існуючих рішень із функціями динамічного балансування варто виділити Ganglia [11] та Nagios з розширенням DNX [12]. Проте система Ganglia орієнтована переважно на високопродуктивні обчислювальні кластери та Grid-системи, через що обмежена у зборі даних із мережевого обладнання. Своєю чергою, механізм балансування DNX, побудований за моделлю «брокер-

агент», не враховує топологічні особливості розміщення агентів у мережевій інфраструктурі.

Алгоритми динамічного балансування навантаження, застосовані у розподіленій системі моніторингу, базуються на принципах функціонування Grid-мереж [13, 14]. Згідно з підходом, описаним у [13], модель Grid реалізована як ієрархічна структура, що складається з чотирьох рівнів: Grid, кластерів, вузлів та обчислювальних елементів. Нижній рівень охоплює фізичне обладнання системи, тоді як вищі рівні представлені віртуальними менеджерами ресурсів, що функціонують на базі вузлів кластера.

Балансування навантаження здійснюється ієрархічно: менеджер вузла виконує розподіл між обчислювальними елементами, менеджер кластера - між вузлами, а менеджер Grid - між кластерами. Пріоритетним є внутрішньовузлове балансування. Внутрішньокластерне перерозподілення завдань активується лише у разі неспроможності вузла самостійно оптимізувати навантаження. Міжкластерне балансування застосовується як крайній захід, коли більшість менеджерів вузлів не здатні виконати локальну оптимізацію. Оскільки міграція завдань між кластерами супроводжується значними накладними витратами, порогові значення дисбалансу для цього рівня мають бути встановлені на високому рівні. Перевагою даного підходу є мінімізація інтенсивності обміну повідомленнями за рахунок пріоритетного виконання локальних операцій, при цьому інформація про ресурси агрегується від нижніх рівнів до верхніх.

У моделі [13] менеджер Grid-мережі акумулює відомості про стан системи, виконуючи функції глобального координатора. Попри низьку частоту звернень, цей елемент створює ризик появи «вузького місця» та є потенційною точкою відмови. Для усунення вказаних недоліків у роботі [14] запропоновано повністю розподілену модель, де відсутній центральний вузол (корінь ієрархії). Натомість менеджери кластерів діють як рівноправні суб'єкти, що самостійно здійснюють як внутрішньокластерне, так і міжкластерне балансування. Менеджер кластера відповідає за моніторинг завантаженості вузлів, обчислення робочого навантаження, прийняття рішень щодо балансування та координацію взаємодії з іншими менеджерами. Вузол, своєю чергою, забезпечує оновлення даних про власне навантаження та виконання відповідних директив менеджера. Модель [14] передбачає виділення окремого вузла для функціонування менеджера кластера, проте допускається і варіант дуального використання вузла - як для виконання обчислювальних завдань, так і для адміністрування.

Проте видалення кореня дерева збільшує кількість повідомлень, що передаються між менеджерами кластерів: замість передачі одного інформаційного повідомлення про завантаження менеджеру мережі кожен менеджер кластера надсилає повідомлення іншим менеджерам кластерів.

У роботі [14] обґрунтовано, що використання розподіленого підходу до балансування системи забезпечує суттєвіші переваги в часі відгуку порівняно з ієрархічним методом. Встановлено, що при незмінній кількості кластерів та зростанні обсягу завдань перевага розподіленого підходу збільшується поступово, тоді як ієрархічний підхід демонструє вищу динаміку зростання ефективності до моменту, коли мережевий менеджер Grid стає вузьким місцем системи.

У працях [15, 16] ці результати адаптовано для цілей балансування розподіленої системи моніторингу, що функціонує на базі вільних ресурсів наявних серверів (серверів-агентів). Моніторинг здійснюється шляхом періодичного опитування стану мережі за допомогою протоколу SNMP, при цьому кожен параметр MIB (Management Information Base) розглядається як окреме джерело даних.

Під час розробки алгоритмів балансування навантаження пріоритетними завданнями є мінімізація додаткового навантаження на сервери-агенти та оптимізація обсягів передачі даних у системі моніторингу. Надмірний трафік виникає у випадках, коли джерела даних і сервер-агент, що виконує їх опитування, територіально або логічно розділені.

Для вирішення завдання динамічного балансування навантаження в розподіленій системі моніторингу сервери-агенти об’єднуються у логічну деревоподібну структуру. У загальному випадку кількість рівнів такої ієрархії не є обмеженою та визначається загальною кількістю агентів у системі [13].

Застосуємо наступні позначення: $S = \{sk\}$ - кількість серверів мережі; SA – сервери мережі, на яких розміщені агенти системи моніторингу, $SA \subseteq S$, $N_{SA} = |SA|$; $D = \{dk\}$ - кількість джерел даних, $N_D = |D|$; DS_m - джерела даних, призначених для зберігання m-го агента, $\bigcup_{m=1}^{N_{SA}} DS_m = D$, $\bigcap_{m=1}^{N_{SA}} DS_m = \emptyset$; DL_m -

множина джерел даних, що опитуються m-м агентом, $\bigcup_{m=1}^{N_{SA}} D\mathrm{L}_m = D$, $\bigcap_{m=1}^{N_{SA}} DL_m = \emptyset$.

Сервери-агенти функціонують як дуальні вузли, що поєднують виконання прикладних завдань мережевого сервера з функціями управління та опитування джерел даних. Оскільки обсяг робочого навантаження агентів є динамічним, це безпосередньо впливає на доступні ресурси для підготовки та проведення опитувань. Унаслідок цього множини призначених агенту та опитуваних ним джерел, як правило, не збігаються ($DL_m \neq DS_m$). Для оптимізації процесів перед кожним циклом опитування застосовується механізм динамічного балансування, що дозволяє актуалізувати перелік джерел даних для кожного агента [14].

Динамічне балансування навантаження здійснюється у два етапи. Перший етап передбачає локальне балансування, що виконується кожним агентом паралельно та незалежно від інших. У межах цього процесу агент оцінює спроможність опрацювання закріплених за ним джерел даних. Відтак, на етапі локального балансування агенти керуються таким критерієм:

$$\left|DS_m \cap DL_m\right| \to \max$$

також враховуються обмеження по кількості доступних ресурсів:

$$Z_i \cdot \left|DL_m\right| \le R_{mi},\ \ m = \overline{1, N_{SA}},\ m = \overline{1,3},$$

де Z_i - кількість ресурсів i-го типу, необхідних для опитування джерела даних;

R_{mi} - кількість доступних ресурсів i-го типу m-го агента до початку виконання локального балансування.

Передбачається, що кожен сервер володіє чотирма типами ресурсів: процесорними, оперативною пам’яттю, мережевими та дисковими, що відповідають індексам i = 1, 2, 3, 4.

Попередній аналіз свідчить про доцільність виключення ресурсів зовнішньої пам'яті ($i = 4$) із розгляду в межах цих алгоритмів.

Припускається, що всі джерела даних є ідентичними за обсягом необхідних ресурсів. Водночас ці вимоги підлягають масштабуванню для кожного сервера з урахуванням його продуктивності, місткості та пропускної здатності.

Приймемо DS'_m – як підмножину власних джерел даних, опитування яких зможе виконати m-й агент, тобто $DS'_m \subseteq DS_m$; а DP_m - підмножина власних джерел даних, яка буде передана батьківському агенту для опитування іншими агентами під час виконання другого етапу багаторівневого балансування, $DS_m = DS'_m \bigcup DP_m$; DC_m - множина джерел даних, які будуть передані для опитування агенту s_m на другому етапі балансування. Нехай також SC_m - множина дочірніх агентів у агента s_m у логічній (ієрархічній) структурі агентів, а fs_m - значення прапора синхронізації для m-го агента.

У процесі балансування m-й агент ($m = \overline{1, N_{SA}}$) виконає вісім кроків алгоритму [14].

1. Визначає максимальну кількість джерел даних, які він може опитати, $N_{ds\max m} = \min\limits_{i=\overline{1,3}} \frac{R_{mi}}{Z_i}$, а потім встановлює $DL_m = \emptyset$.

2. Перевіряє виконання умови $N_{ds\max m} \geq |DS_m|$. Якщо умова не виконується, переходить до кроку 4.

3. У агента достатньо ресурсів для опитування всіх своїх джерел даних, тобто $DS'_m = DS_m$. Перехід до кроку 5.

4. Агент не має достатньо ресурсів, щоб опитати всі розміщені на ньому джерела. Агент формує довільну підмножину DS'_m розмірністю $N_{ds\max m}$ з множини призначених йому для зберігання джерел, тобто $DS'_m \subset DS_m$, $|DS'_m| = N_{ds\max m}$. Решта джерел утворюють множину $DP_m = DS_m \setminus DS'_m$, що містить $|DS_m| - N_{ds\max m}$ джерел.

5. Множина DS'_m включається до рішення DL_m: $DL_m = DS'_m$.

6. Агент приступає до опитування розподілених йому джерел даних (множини DS'_m).

7. Агент оновлює кількість ресурсів, що у нього залишилися:

$$R'_{mi} = R_{mi} - Z_i \cdot |DS'_m|,\ i = \overline{1,3}$$
$$R_{mi} = R'_{mi},\ i = \overline{1,3}.$$

8. Якщо у агента немає нащадків (він є кінцевою вершиною дерева агентів), він надсилає батьківському агенту множини DP_m та $\{R_{mi} | i = \overline{1,3}\}$ і встановлює прапорець синхронізації fs_m = 1. Виконання першого етапу балансування завершується.

У межах логічного дерева множину агентів доцільно класифікувати на термінальні вершини та вузли. Своєю чергою, множину вузлів структуровано на підмножини ST^i, I = 1, 2, …, які відповідають ієрархічним рівням логічного дерева.

Другий етап динамічного балансування - багаторівневе балансування - ініціюється паралельно агентами s_m множини ST^1. Для цих агентів усі нащадки є термінальними вершинами логічного дерева, що відповідає умові $\forall s_m$, $s_m \in ST^1$, якщо $\forall n(s_n \in SC_m) \to fs_n = 1$. Багаторівневе балансування реалізується через ітеративне виконання базових дворівневих процедур, кожна з яких охоплює піддерево, що містить відповідного агента та всі його дочірні вузли чи вершини. Зокрема, піддеревом агента s_m є множина $SD_m = \{s_m\} \cup SC_m$. Балансування піддерев активується автоматично для кожного агента, усі нащадки якого мають встановлений прапорець синхронізації (рівний 1). На початковому етапі багаторівневого балансування прапорці синхронізації встановлено виключно для термінальних вершин дерева. Надалі цей стан поширюється на вузли, піддерева яких завершили процедуру балансування. За умови ігнорування варіативності швидкості виконання операцій на різних піддеревах, результатом багаторівневого балансування стає послідовне встановлення прапорців синхронізації на агентах-вузлах множин ST^1, потім ST^2 тощо.

Розглянемо алгоритм балансування піддерева, що виконується агентом $s_m \in ST^i$ i-го рівня дерева, i = 1, 2, ….

1. Для агентів піддерева $SD_m = \{s_m\} \bigcup SC_m$ агент s_m визначає максимальну кількість джерел даних, які може опитати кожен з агентів піддерева: $\forall n$, $s_n \in SD_m$:

$$N_{ds\max n} = \min_{i=\overline{1,3}} \frac{R_{ni}}{Z_i},$$

де R_{ni} - кількість доступних ресурсів *i*-го типу *n*-го агента після виконання локального балансування.

2. Для кожного агента множини SD_m створюється порожня множина $DC_n\left(\forall n, s_n \in SD_m \to DC_n = \emptyset\right)$, яка міститиме джерела даних, призначені для опитування відповідного агента на даному етапі.

3. Агенти множини SD_m сортуються за спаданням величини $N_{ds\max n}$. Відсортована множина має вигляд

$$SD_m = \left\{ s_k \middle| \forall k (k < |SD_m|) N_{ds\max k} > N_{ds\max k+1} \right\}.$$

4. Множини DC_n заповнюються нерозподіленими джерелами даних відповідно до значень максимальної кількості джерел даних $N_{ds\max n}$, які можна призначити агенту для опитування.

Алгоритм призначення джерел даних для опитування базується на введенні дискретних рівнів l = 1, 2, ... для величин $N_{ds\max n}$, причому l = 1 відповідатиме найбільшому значенню $N_{ds\max n}$ (рис. 3.1). Якщо з'являються агенти з однаковими значеннями $N_{ds\max n}$, вони об'єднуються в один рівень, тобто $l = \overline{1, l_{\max}}$, $l_{\max} \le |SD_m|$. Позначимо значення величини $N_{ds\max n}$, яке відповідає рівню l, як $N_{lev\,l}$

Позначимо через N_l максимальну кількість джерел даних, що додаються на рівні *l* до кожної з множин DC_n, для яких $N_{ds\max n} \ge N_{lev\,l}$:

$$\forall l, l < l_{\max} \to N_l = N_{lev\,l} - N_{lev\,l+1};$$
$$l = l_{\max} \to N_{l\max} = N_{lev\,l\max}.$$

Припустимо $DM_m = \bigcup_{s_n \in SC_m} DP_n \cup DP_m$ – множина нерозподілених джерел даних для піддерева $SD_m = \{s_m\} \cup SC_m$, які, можуть бути перерозподілені між вузлами даного піддерева. Копію цієї множини позначимо як $DM'_m = DM_m$, де $DM'_m = \{d'_k\}$.

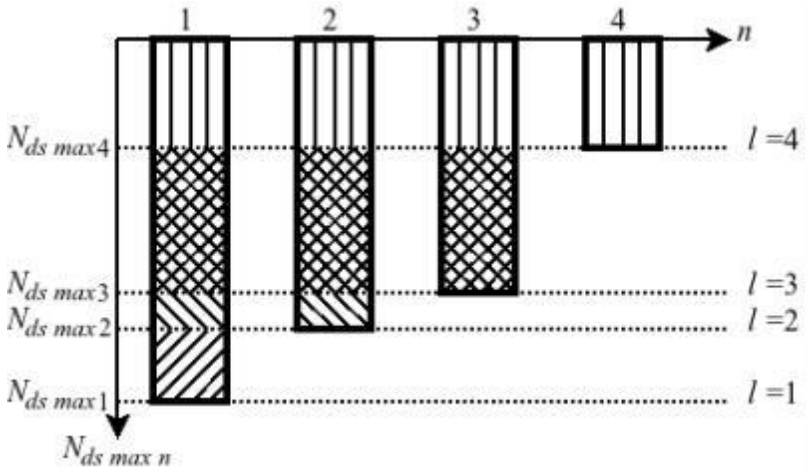

Рисунок 3.1 - Розподіл джерел даних за рівнями

Якщо значення $N_{ds\max n}$ у агента більше чи рівне значенню $N_{lev\ l}$ (агент знаходиться на рівні l), то йому призначаються для опитування N_l довільних джерел даних із множини DM'_m, які входять до множини DC_n. Умова завершення даного кроку: $DM'_m = \emptyset$ або $DM'_m \neq \emptyset, l = l_{\max}$.

Наприклад, на рис. 3.1 на першому рівні агенту s_1 буде призначено для опитування ($N_{ds\max 1} - N_{ds\max 2}$) джерел даних, на другому - агентам s_1, s_2 - ($N_{ds\max 3} - N_{ds\max 2}$) джерел даних тощо.

5. Множини DC_n включаються в рішення:

$$DL'_n = DC_n \cup DL_n,\ DL_n = DL'_n,\ s_n \in SD_m.$$

У результаті досягається рівномірний розподіл навантаження на агентів, при якому пріоритет надається тим, хто володіє найбільшим обсягом ресурсів. Таким чином, цей алгоритм забезпечує максимізацію мінімального запасу ресурсів:

$$\min_{m,i}\left(R_{mi} - Z_i \cdot \left|DL_m\right|\right) \to \max,\ i = \overline{1,3},\ m = \overline{1, N_{SA}}.$$

6. У разі неможливості розподілу джерел даних між нащадками, вони об'єднуються у множину для подальшої передачі батьківському вузлу. Для реалізації цього процесу агент s_m спочатку ініціалізує порожню множину $DP_m = \emptyset$, після чого здійснює додавання до неї відповідних джерел даних:

$$DP_m = DM_m \setminus \bigcup_{\forall n, s_n \in SD_m} DC_n.$$

7. Після чого проводиться підрахунок ресурсів, що залишилися:

$$R'_{ni} = R_{ni} - Z_i \cdot \left|DC_n\right|,\ R_{ni} = R'_{ni},\ i = \overline{1,3},\ s_n \in SD_m$$

8. Агент надсилає нащадкам множини джерел даних для опитування $DC_n\left(\forall n, s_n \in SD_m\right)$.

9. Агент s_m пересилає батьківському агенту, що належить множині ST^{i+1}, сформовані в результаті балансування піддерева множини DP_m та $\left\{R_{mi} \middle| i = \overline{1,3}\right\}$.

10. Агент s_m встановлює прапор синхронізації $fs_m = 1$.

Алгоритм дворівневого балансування вважається завершеним після завершення балансування піддерева кореневого вузла-агента. Якщо за результатами виконання виявлено, що $DP_m = \emptyset$, процес балансування навантаження вважається успішно завершеним. У разі, якщо $DP_m \neq \emptyset$, кореневий агент дерева ініціює процедуру глобального багаторівневого балансування. У межах цієї процедури здійснюється послідовний аналіз резервів ресурсів кожного агента системи моніторингу з подальшим призначенням нерозподілених джерел опитування тим агентам, у яких було зафіксовано вільні ресурси.

У разі, якщо після завершення глобального багаторівневого балансування залишаються нерозподілені джерела, необхідно здійснити розгортання нових

агентів на базі найменш завантажених серверів корпоративної мережі. Слід зазначити, що виконання глобального багаторівневого балансування зумовлює потребу в повторному вирішенні завдання розподілу джерел даних між агентами для їх подальшого зберігання.

Для оцінки ефективності алгоритму локального балансування було розроблено програмне забезпечення, що реалізує відповідні механізми та містить інтегровані функції моніторингу споживання віртуальної пам’яті та ресурсів процесора. Вимірювання вказаних параметрів здійснювалося шляхом зчитування даних із системного файлу `/proc/[pid]/stat`, де `[pid]` відповідає ідентифікатору процесу локального балансування. Результати проведених досліджень представлені на рис. 3.2 та 3.3.

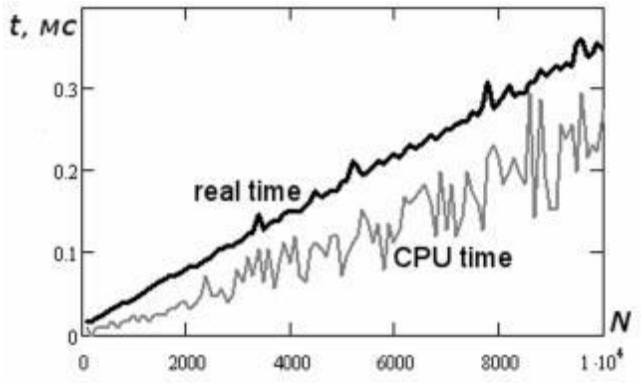

Рисунок 3.2 - Тривалість роботи алгоритму локального балансування

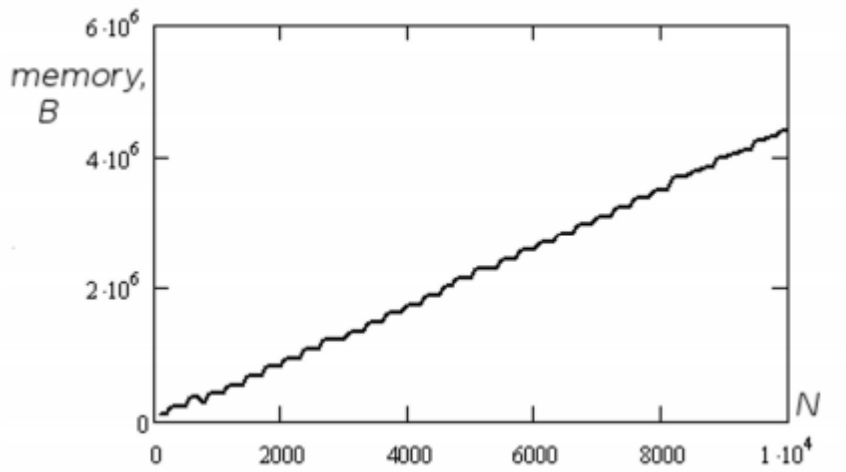

Рисунок 3.3 - Споживання пам'яті під час роботи алгоритму локального балансування

Діапазон кількості джерел даних, обраних для зберігання в обох сценаріях, варіювався від 100 до 10 000 із кроком 100. Як показник споживання процесорних ресурсів було використано кількість тактів (clock ticks), під час яких досліджуваний процес займав ядро процесора в режимі користувача (user mode) або в режимі ядра (kernel mode). Варто зазначити, що тривалість такту залежить від версії ядра операційної системи Linux та її дистрибутива. Кількість тактів на секунду визначається константою конфігурації ядра `CONFIG_HZ`, яка для досліджуваної системи становила 250 (тривалість одного такту - 4 мс). Частота процесорного ядра досліджуваної системи - 3 ГГц, кількість ядер - 1.

На графіку «real time» (рис. 3.2) відображено загальний час виконання алгоритму, тоді як «CPU time» демонструє час безпосереднього використання процесора. Як свідчать дані, наведені на рис. 3.2, залежність часу використання процесора від кількості джерел даних є практично лінійною і описується рівнянням: $t = (3{,}599 \cdot 10^{-5})N$, де t - тривалість завантаження ядра процесора алгоритмом, а N - кількість джерел даних.

Витрати оперативної пам’яті, необхідні для формування тимчасових таблиць із джерелами даних, також демонструють лінійну залежність від кількості оброблюваних об'єктів (рис. 3.3). Рівняння залежності становить $q = 436{,}63N$, де q - обсяг віртуальної пам’яті (у байтах), необхідний для роботи алгоритму локального балансування.

Аналогічні дослідження були проведені також для алгоритму багаторівневого балансування. Результати підтвердили достатню ефективність даного підходу, зважаючи на показники продуктивності та обсяг використання оперативної пам’яті, що відповідає вимогам до системи моніторингу.

4 РЕАЛІЗАЦІЯ МЕТОДУ БАГАТОРІВНЕВОГО БАЛАНСУВАННЯ В КОРПОРАТИВНІЙ МЕРЕЖІ

4.1 Вибір середовища для реалізації методу

Для реалізації та тестування пропонованого методу багаторівневого балансування трафіком із використанням прогнозного моделювання було обрано хмарну платформу Google Colaboratory (Google Colab). Ця інтерактивна інфраструктура забезпечує широкі можливості середовища виконання Python, що дозволяє ефективно розробляти, тестувати та візуалізувати аналітичні моделі в режимі реального часу.

Ключовою перевагою Google Colab є відсутність необхідності в локальному розгортанні програмного забезпечення, що забезпечує незалежність від апаратних обмежень платформи та оптимізує доступ до ресурсомістких обчислень. Середовище інтегроване з актуальними версіями спеціалізованих бібліотек для аналізу даних та машинного навчання (pandas, numpy, scikit-learn, matplotlib, seaborn), що дозволяє реалізувати повний цикл обробки даних - від формування вхідних параметрів до розробки інтелектуальних систем прийняття рішень [3].

Використання апаратного прискорення на графічних (GPU) та тензорних (TPU) процесорах дозволяє масштабувати обчислення при роботі зі складними архітектурами штучного інтелекту, зокрема згортковими та рекурентними нейронними мережами. Інтеграція з хмарним сховищем Google Drive забезпечує структуроване зберігання даних, архівування результатів експериментів та сприяє ефективній спільній роботі над проєктом.

Високий рівень інтерактивності середовища, реалізований через поетапне виконання коду та підтримку розмітки Markdown, сприяє належному документуванню алгоритмів та візуальній інтерпретації отриманих результатів. Такий підхід підвищує прозорість дослідження та спрощує підготовку матеріалів для наукових публікацій чи технічної звітності [3].

Отже, обрання Google Colab як інструментарію для реалізації методу є обґрунтованим завдяки його технічній ефективності, доступності та високому рівню сумісності з сучасними інструментами машинного навчання, що відповідає актуальним стандартам наукових досліджень.

4.2 Реалізація пропонованого методу в Google Colab

Для реалізації зазначеного алгоритму пропонується виконати наступні етапи:

- імпортувати необхідні бібліотеки;
- завантажити або згенерувати набір синтетичних даних трафіку;
- змоделювати трафік у вигляді часового ряду $\lambda(t)$ з варіативними піковими значеннями;
- виконати розмітку даних відповідно до класів трафіку;
- розробити та навчити модель прогнозування;
- візуалізувати отримані результати;
- сформулювати висновки щодо ефективності запропонованого підходу.

На рис. 4.1 представлено фрагмент реалізації механізму адаптивного керування трафіком, що базується на прогнозуванні та балансуванні навантаження. У програмній реалізації встановлено граничне значення пропускної здатності мережі $\mu=130$, яке використовується як пороговий показник для прийняття рішень щодо обслуговування трафіку різних класів. З метою оптимізації розподілу ресурсів прогнозована інтенсивність $\lambda'(t+1)$ порівнюється з відповідними відсотковими частками μ.

Залежно від результатів прогнозування система визначає поточний стан керування:

- за умови, що прогноз становить менше ніж 70% від граничного рівня, дозволяється обслуговування всіх класів трафіку («Усі класи активні»);
- якщо навантаження знаходиться в діапазоні від 70% до 100% від граничного рівня, застосовується політика обмеження фонового трафіку («Фоновий обмежено»);
- у разі перевищення граничного значення μ активується режим пріоритетного обслуговування критичного трафіку («Критичний пріоритет»).

Python

```
threshold = 130
result = []

for real, pred in zip(y_test, y_pred):
    if pred < 0.7 * threshold:
        result.append('Усі класи активні')
    elif pred < threshold:
        result.append('Фоновий обмежено')
    else:
        result.append('Критичний пріоритет')

df_result = pd.DataFrame({
    'Прогноз': y_pred,
    'Стан': result
})

df_result['Стан'].value_counts()
```

Результат виконання (Output)

Стан	count
Фоновий обмежено	179
Критичний пріоритет	19

Рисунок 4.1 – Реалізація адаптивного методу балансування навантаженням

У нижній частині рисунка наведено розподіл прогнозованих станів. Згідно з отриманими даними, у 179 випадках спостерігалося помірне перевантаження, що зумовило обмеження фонового трафіку. У 19 випадках зафіксовано критичне перевищення навантаження, що вимагало переходу до високопріоритетного режиму обслуговування. Відсутність стану «Усі класи активні» свідчить про високу інтенсивність трафіку протягом усього періоду моніторингу, що є характерним для експлуатації системи в умовах постійного високого навантаження.

Зазначений підхід демонструє ефективність використання прогнозних моделей для адаптивного керування ресурсами мережі як у реальному, так і в симульованому середовищі. Отримані результати можуть бути використані для оцінки стійкості політики обслуговування до коливань трафіку та подальшого вдосконалення моделей пріоритезації.

Вихідний код програмної реалізації наведено в додатку А.

4.3 Аналіз результатів дослідження

На рис. 4.2 представлено динаміку зміни інтенсивності чотирьох типів трафіку в корпоративній мережі протягом досліджуваного періоду. Графік побудовано на основі синтетичних даних, що моделюють характеристики передачі голосового, відео-, інформаційного та фонового потоків. Аналіз свідчить про значну варіативність відеотрафіку, для якого характерні періодичні пікові навантаження, тоді як голосовий трафік демонструє стабільність із мінімальними коливаннями. Інформаційний трафік характеризується помірною інтенсивністю з епізодичними відхиленнями, а фоновий трафік — хвилеподібною динамікою з вираженою сезонністю. Представлена візуалізація дозволяє оцінити нерівномірність розподілу навантаження між типами трафіку та слугує аналітичним підґрунтям для розробки політик пріоритезації в межах системи адаптивного управління.

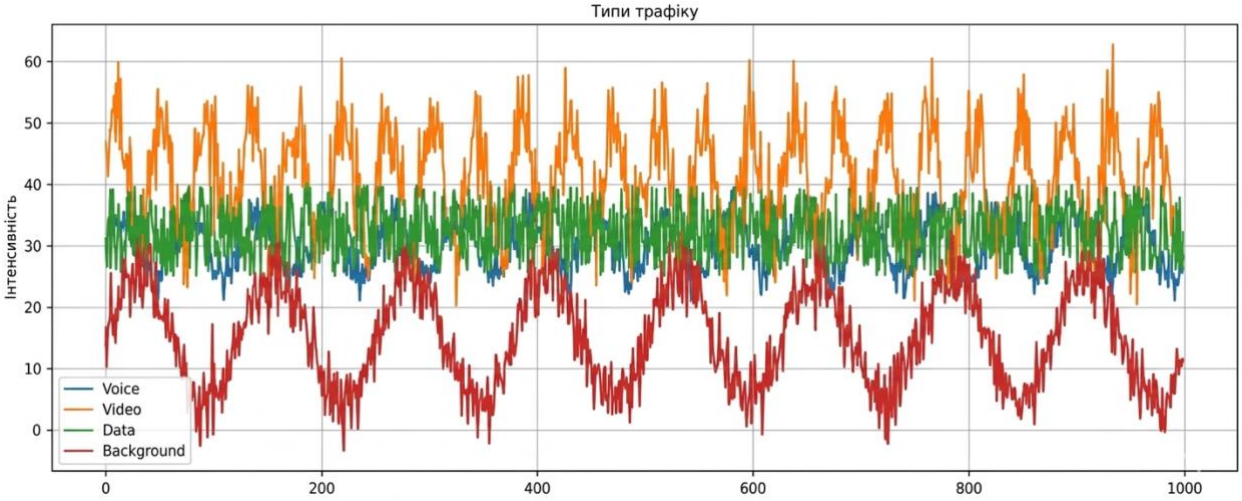

Рисунок 4.2 – Сгенеровані типи трафіку

На рис. 4.3 наведено порівняльний аналіз фактичних та прогнозованих показників сумарного мережевого трафіку (Total traffic) за визначений період спостереження. Синя лінія відображає реальні значення інтенсивності навантаження, тоді як помаранчева демонструє результати прогнозування, отримані за допомогою розробленої моделі. Високий ступінь кореляції між прогнозованими даними та фактичними коливаннями трафіку підтверджує задовільну точність моделі у відтворенні загальної динаміки та тенденцій.

Водночас виявлені відхилення у пікових зонах вказують на необхідність подальшої оптимізації моделі з метою підвищення її чутливості до різких змін інтенсивності навантаження. Представлена візуалізація є критично важливою для верифікації ефективності прогнозних алгоритмів у контексті впровадження систем адаптивного управління мережевими ресурсами.

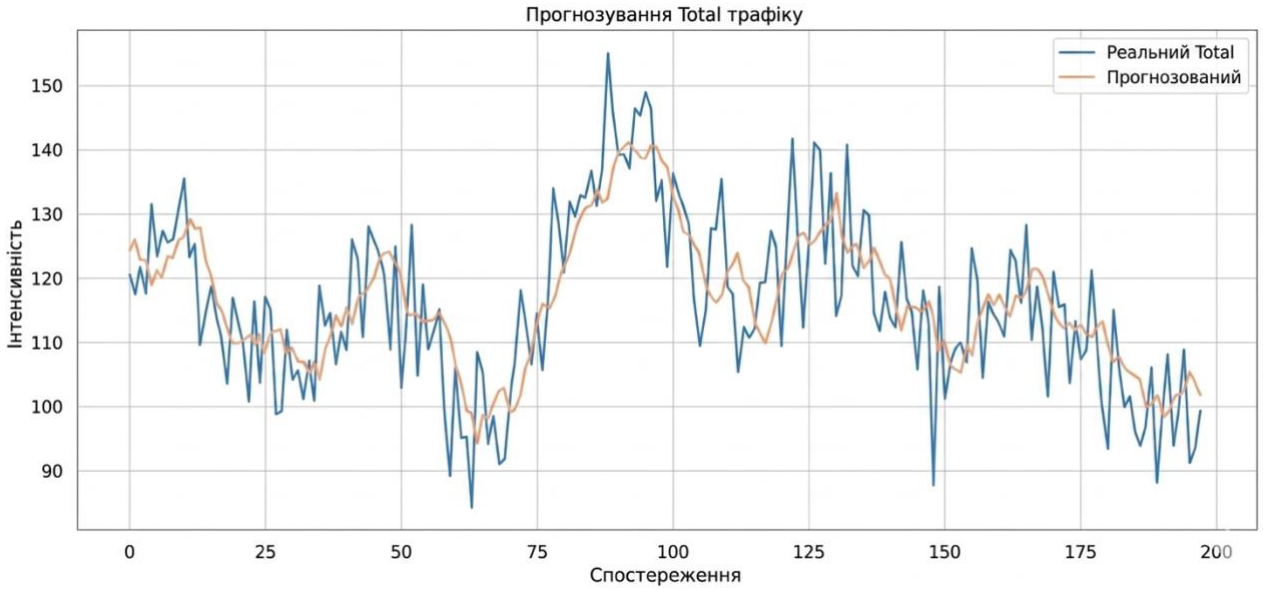

Рисунок 4.3 – Прогнозування загального трафіку

На рис. 4.4 відображено гістограму розподілу прогнозованої інтенсивності трафіку, сформовану за результатами застосування методу динамічного балансування навантаженням та згруповану за типами прийнятих рішень. Графік демонструє кореляцію між діапазонами прогнозованих значень і відповідними станами системи: «Фоновий обмежено» та «Критичний пріоритет».

Як свідчать отримані дані, основний масив прогнозів зосереджений у діапазоні помірного навантаження (105–125 одиниць), що ініціює реалізацію політики часткового обмеження трафіку низького пріоритету. Збільшення частоти випадків у діапазоні понад 130 одиниць вказує на перехід системи до критичного режиму функціонування, за якого пріоритет надається виключно трафіку найвищого рівня. Проведений аналіз забезпечує можливість кількісної оцінки частоти перевищення гранично допустимих значень та становить обґрунтування для впровадження механізмів динамічного регулювання мережевого навантаження в умовах прогнозованої нестабільності.

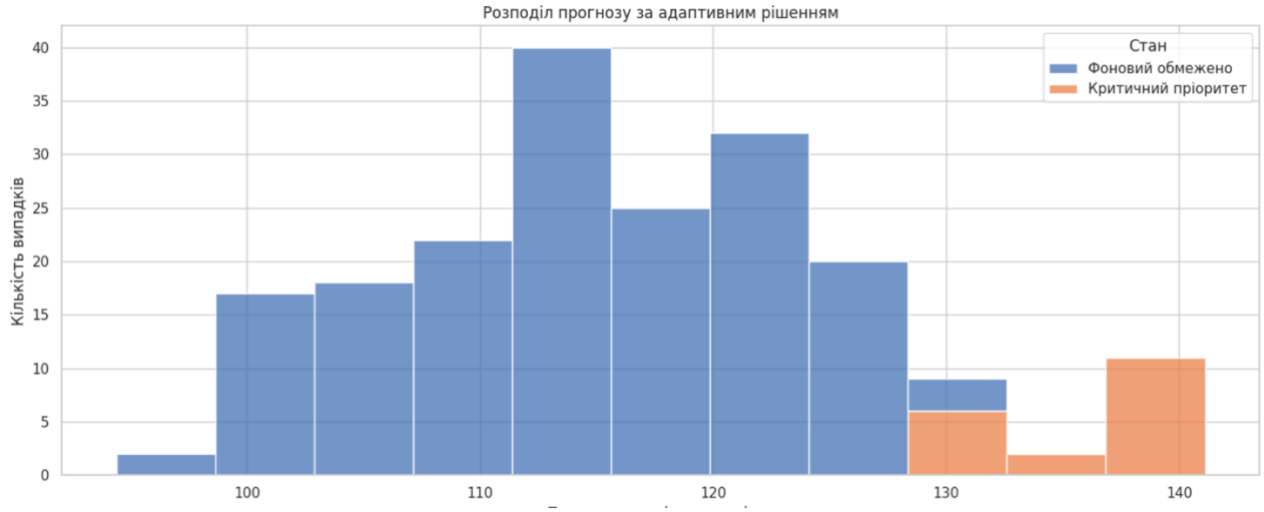

Рисунок 4.4 – Розподіл прогнозу за адаптивним рішенням

На рисунках 4.5 та 4.6 наведено результати порівняльного аналізу ефективності трьох моделей машинного навчання, застосованих для прогнозування інтенсивності мережевого трафіку в межах методу динамічного балансування навантаженням.

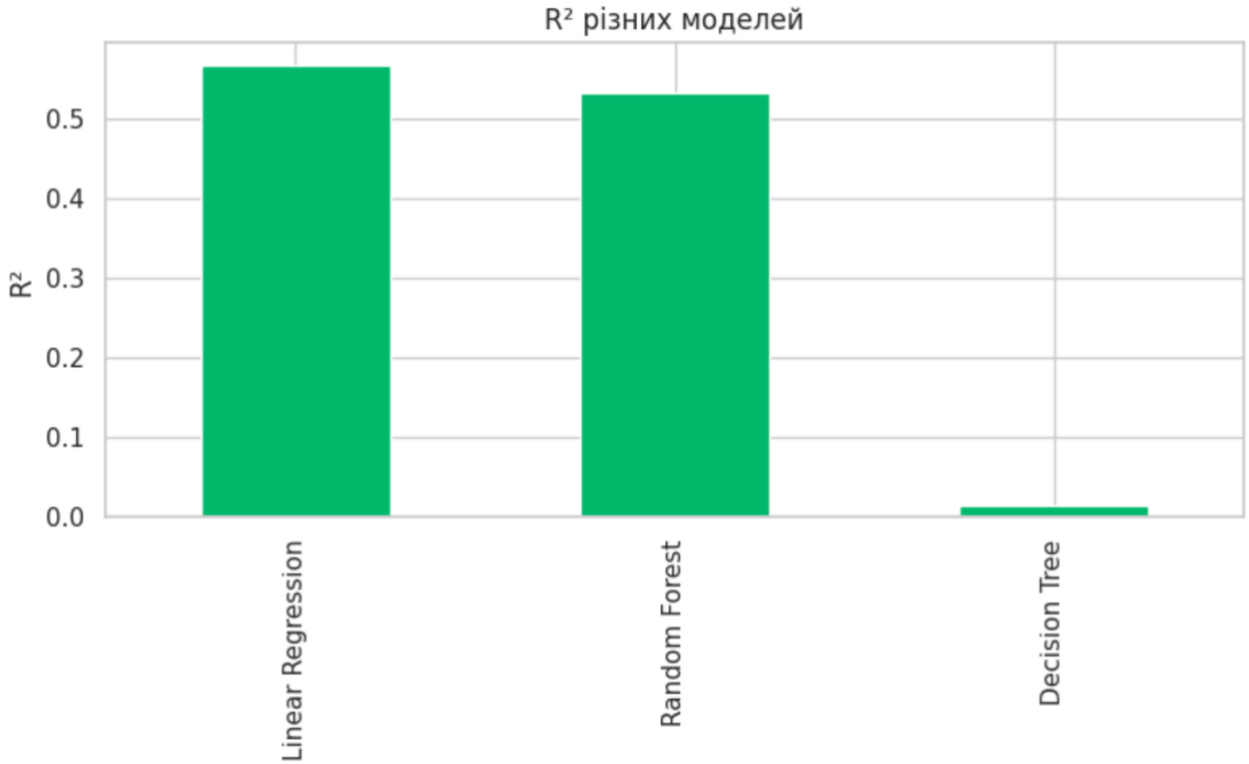

Рисунок 4.5 – Графік порівняння моделей за коефіцієнтом детермінації (R^2)

На першому графіку відображено середньоквадратичну помилку (MSE), що характеризує кількісну точність моделей. Найнижчий показник помилки

зафіксовано у моделі лінійної регресії. Дещо вищий рівень похибки демонструє модель випадкового лісу (Random Forest), тоді як дерево рішень (Decision Tree) виявилося найменш ефективним, показавши суттєво вищий рівень відхилень.

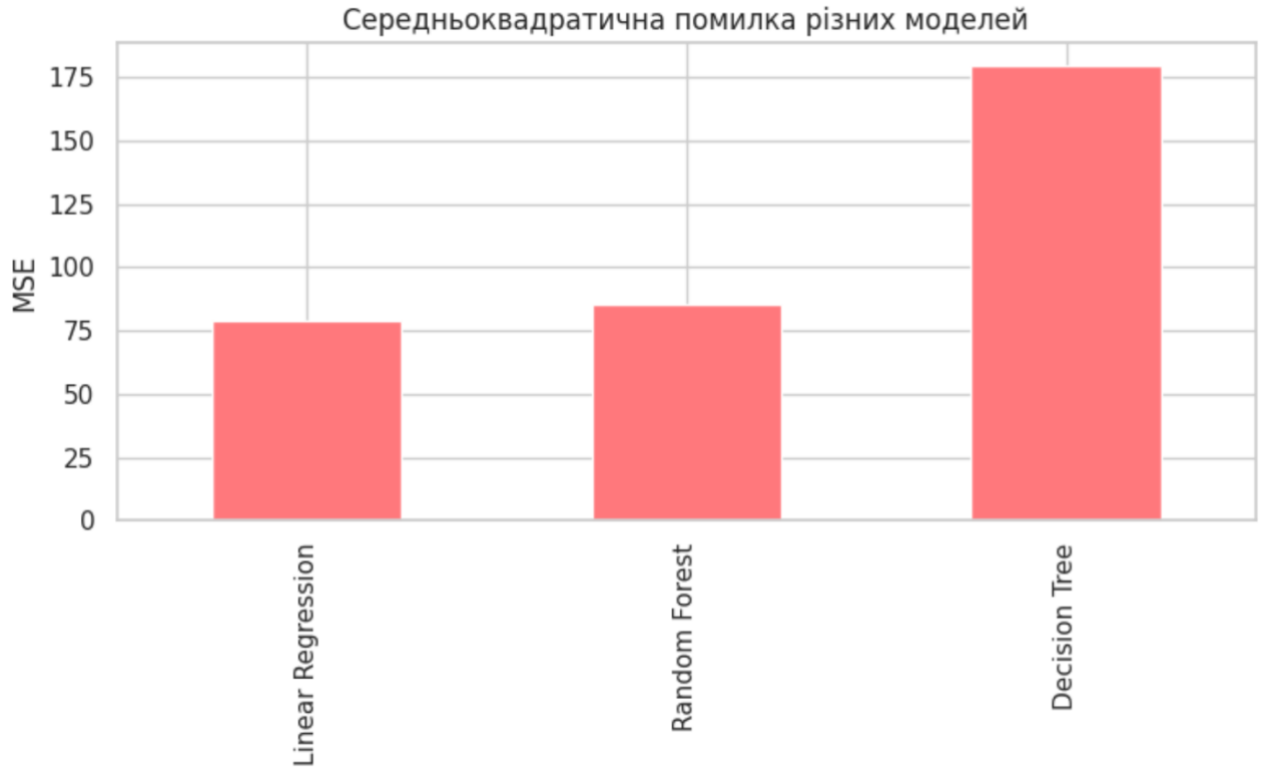

Рисунок 4.6 – Графік порівняння моделей за точністю (MSE)

Другий графік ілюструє коефіцієнт детермінації (R^2), який визначає якість апроксимації залежності між вхідними ознаками та цільовим показником. Найвищу відповідність між прогнозованими та фактичними даними знову забезпечує лінійна регресія, що підтверджує її доцільність для задач із переважно лінійною структурою даних. Модель Random Forest демонструє прийнятний рівень узгодженості, натомість дерево рішень має мінімальне значення R^2, що вказує на низьку здатність до генералізації та підвищену варіативність результатів прогнозування.

Результати дослідження свідчать, що для прогнозування навантаження в корпоративній мережі доцільно застосовувати інтерпретовані моделі або ансамблеві методи, які демонструють вищу узгодженість із характером вхідних даних. Отримані метрики є вагомим підґрунтям для обґрунтування вибору інструментарію та підтверджують валідність запропонованого методу.

Прогнозована інтенсивність сумарного трафіку (Total), згідно з графічними даними, варіюється в межах 90-150 одиниць, що зумовлює такі статистичні характеристики:

- розмах значень становить близько 60 одиниць;
- типова варіація (стандартне відхилення) становить 10-15 одиниць.

Аналіз результатів моделювання виявив наступне:

1. Лінійна регресія та Random Forest: MSE становить 80-90, що відповідає середньоквадратичній похибці близько 9-10 одиниць. Враховуючи специфіку даних, такий рівень похибки є прийнятним. Моделі демонструють здатність адекватно відтворювати загальні тренди, навіть за наявності шумів.

2. Дерева рішень (Decision Tree): MSE перевищує 170, що свідчить про високу похибку відносно масштабу даних, ймовірно, внаслідок перенавчання або недостатньої здатності моделі до узагальнення.

Отже, показники MSE у діапазоні 80-90 є достатніми для прогнозування трафіку в реальних або симульованих системах. Це підтверджує ефективність розробленого методу для цілей динамічного керування мережевими ресурсами.

ВИСНОВКИ

У межах кваліфікаційної роботи проведено комплексне дослідження, аналіз та практичну реалізацію сучасних методів і моделей управління трафіком у корпоративних мережевих інфраструктурах. Робота охоплює теоретичні засади функціонування механізмів контролю потоків даних, а також прикладну складову, що включає моделювання, прогнозування та оцінку ефективності запропонованого підходу на базі алгоритмів машинного навчання.

Проведений аналіз засвідчив, що традиційні методи управління (статистичні, динамічні та евристичні) характеризуються обмеженою адаптивністю до стрімких змін мережевого навантаження. Це зумовило необхідність розробки комбінованого підходу, що інтегрує точність аналітичних моделей із гнучкістю штучного інтелекту. У результаті було впроваджено методу динамічного балансування трафіком із застосуванням прогнозування навантаження, що дозволяє оптимізувати розподіл мережевих ресурсів на основі завчасного виявлення пікових станів.

У середовищі Google Colab розроблено повнофункціональну модель, яка забезпечує генерацію синтетичного трафіку, побудову прогнозних моделей, класифікацію пакетів за рівнем критичності та динамічне реагування на зміни параметрів мережі. Порівняльне тестування алгоритмів машинного навчання (лінійна регресія, дерева рішень, випадковий ліс) продемонструвало найвищу ефективність лінійної регресії, яка забезпечила мінімальну середньоквадратичну похибку та високий коефіцієнт детермінації.

Отримані результати підтверджують високу ефективність запропонованого методу в динамічних корпоративних середовищах. Запропонований підхід є придатним для впровадження в реальні мережеві інфраструктури з метою забезпечення стійкості сервісів, підвищення якості обслуговування (QoS) та оптимізації використання пропускної здатності каналів.

Основні положення роботи висвітлено у тезах до конференції 30-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті» [15].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Michael J. Kavis. Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS). 1st ed, Wiley, 2014. 224 p.

2. Flach P.A. Machine Learning: The Art and Science of Algoritms that Makes Sense of Data. Cambridge: Cambridge University Press, 2012. 291 p. https://doi.org/10.1017/CBO9780511973000

3. Rolf Oppliger. Ssl and Tls: Theory and Practice.3rd ed, Artech House, 2023. 388 p.

4. Jean-Georges Perrin. Spark in Action. 2nd ed, Manning, 2020. 576 p.

5. Jack G Nestell, David L Olson Professor. Successful ERP Systems: A Guide for Businesses and Executives. Business Expert Press, 2017. 134 p.

6. Brij Kishore Pandey, Emily Ro Schoof. Building ETL Pipelines with Python: Create and deploy enterprise-ready ETL pipelines by employing modern methods. 1st ed, Packt Publishing, 2023. 246 p.

7. Edward Pollack. Analytics Optimization with Columnstore Indexes in Microsoft SQL Server: Optimizing OLAP Workloads. 1st ed, Apress, 2022. 300 p

8. Dr. Logan Song. The Self-Taught Cloud Computing Engineer: A comprehensive professional study guide to AWS, Azure, and GCP. 1st ed., Packt Publishing, 2023. 472 p.

9. Diachenko D., Korobeinikov M., Korobeinikov O., Kovalenko A., Kravchenko P. Data processing methods in a corporate network. Системи управління, навігації та зв'язку, вип.3. Полтава, 2025. С.81-86.

10. Ганчук А.А., Соловйов В.М., Чабаненко Д.М. Методи прогнозування: навч. посіб. / Черкаси: Брама-Україна, 2012. 140 с.

11. Хлапонін Ю. І., Жиров Г. Б., Нікітчін О. М. Застосування нейронних мереж в статистичній системі аналізу і моніторингу телекомунікаційних мереж. Технологічний аудит і резерви виробництва, т. 5, № 2, 2016. С. 35–41.

12. Matthew L. Massie, Brent N. Chun, David E. Culler. The ganglia distributed monitoring system: design, implementation, and experience // Parallel Computing. №30. 2004. P. 817 – 840.

13. Ngu en Thai, D. Real-Time Scheduling in Distributed Systems / D. Nguen Thai // International Conference on Parallel Computing in Electrical Engineering: IEEE Computer Society Press. – Warsaw, Poland, 2002. – P. 165–170.

14. Yagoubi B., Slimani Y. Task Load Balancing Strategy for Grid Computing // Journal of Computer Science 3 (3): Department of Computer Science, Faculty of Sciences. Oran: University of Oran. 2007. P. 186-194.

15. Латишева К.С. Дослідження механізмів керування трафіком у корпоративних мережах // 30-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. Матеріалів форуму. Т.4. / [Електронний ресурс] – Харків: ХНУРЕ. 2026. – 127-128 с.